

## スイングパッド軸受けの動特性

著者	小嶋 裕
学位授与機関	東京商船大学
学位授与年度	1992
URL	<a href="http://id.nii.ac.jp/1342/00000847/">http://id.nii.ac.jp/1342/00000847/</a>

# 学 位 論 文

題 目 スイングパッド軸受の動特性

---

指 導 教 授 岩本 勝美

---

商 船 学 研 究 科 機関学 専 攻

平成 3 年 入 学

氏 名 小嶋 裕

---

平成 5 年 1 月 29 日提出



# 目次

## 第1章 序論

- 1-1 スイングパッド軸受
- 1-2 従来の研究とその問題点
- 1-3 本研究の目的と研究方針

## 第2章 実験装置及び実験方法

- 2-1 実験装置の構成
- 2-2 実験装置概要
  - 2-2-1 供試軸受
  - 2-2-2 軸
  - 2-2-3 ディスク
  - 2-2-4 軸受ブロック
  - 2-2-5 駆動系
  - 2-2-6 給・排油系
  - 2-2-7 測定系
- 2-3 実験方法
  - 2-3-1 非接触変位計の校正
  - 2-3-2 軸の固有振動数測定
  - 2-3-3 熱電対の温度校正
  - 2-3-4 作動流体の粘度と比重の測定
  - 2-3-5 ロータ系の不つり合い除去
  - 2-3-6 測定、観測・記録方法

## 第3章 結果と考察

- 3-1 不つり合い応答
- 3-2 負荷方向の比較
- 3-3 軸受幅による比較
- 3-4 軸重量の違いによる比較
- 3-5 軸受半径隙間の違いによる比較
- 3-6 スイングパッド軸受と真円軸受の比較

## 第4章 結論

## 参考文献

## 付録

# 第一章 序論

## 1-1 スイングパッド軸受の特徴

スイングパッド軸受は、パッド部を複数のゴム層と金属層が積層構造になった支持部によって支持された軸受である。

その軸受は目的に応じ、スラストタイプ、ジャーナルタイプ、パッドの枚数、寸法等、様々な形状がある。（図1-1 参照）

通常、スイングパッド軸受は、パッドの潤滑面と、背面との曲率半径と、その中心は異なっている。

またその形状のため、軸を回転させると、潤滑油の圧力発生でパッド支持部のゴム層が変形し、早い時期にくさび形状の油膜を形成する。

その他に、スイングパッド軸受はその形状のため以下のような利点がある。

浮動ブッシュ軸受、多円弧軸受、ティルティングパッド軸受と同様な振動減衰効果が得られる。

他の軸受と違いパッドの取り付け精度が要求されない。

そのため、ミスアライメントを許容する能力を有する。即ち、ゴミ等を軸と軸受面に噛んでも支持ゴムの変形によって許容される。

またその時、その支持部のゴムの変形によりゴミ等の異物を排除する能力を有する。

弾性支持されていない軸受に比べ軸の低回転時や発停時にも良好な潤滑油のくさび形状を形成する。

多くの利点がある反面以下の様な欠点がある。

同じ偏心率の他の軸受に比べ、この軸受は支持部にゴム層を有するために、軸の回転時に発生する油膜反力により変形するため高い負荷を支持できない。

ゴム層劣化の問題等がある。

前者よりスイングパッド軸受は軸受荷重の軽い水平軸、もしくは垂直軸の支持に適していると考えられる。

## 1 - 2 従来の研究とその問題点

前項のような利点があるにも関わらず、スイングパッド軸受に関する研究は現在まで

ほとんど行われていない。

理論と実験について比較したものは、Y.P.Chiu の行なった研究だけがあるだけである。しかしながら、彼の論文では定常状態の考察のみで、動特性に関する研究は行なわれていない。特に、この軸受に関する安定性、不釣り合い応答などは考察されていない。

彼は、定常状態での 12 枚のパッドを持つスイングパッド軸受についての研究を行った。

その中では、油膜厚さ、発生圧力、パッド部の変形と摩擦係数の解析を行ないっており。その解析を用い、5 種類の形状の軸受を計算し、その結果を比較した。

その時の解析では、彼は 1 パッドをモデル化しており、パッド支持部のゴムの変形はせん断方向にのみ生じると仮定した。

また、半径方向のゴムの変形を考慮に入れるために、ゴムの外側に仮想のバネを考えた。前者の仮定は、ゴムの層の厚さが、パッド寸法に比べ非常に小さいため、軸の接線方向の力による変形より、軸の半径方向の力による圧縮の変形の方が微小になると考えたためである。

以上が、彼の行なった研究の概要である。しかし彼の行なった研究には、以下に書くような問題点がある。

微小と考えた、半径方向の圧縮変形はバネに置き換えて考えたと言うことは、半径方向の変形は当初の予想以上であったと考えられる。そし

て、軸とパッドの間の油膜に発生する圧力分布のため、パッドが軸方向に平行な軸に関する回転方向の移動があることがわかる。彼はこの時のくさび形状の解析を、ティルテイングパッド軸受と同様な方法の解析を行っている。そのため、実際のゴムの変形を正確に解析できていないと考えられる。また彼の行なった解析は静特性についてであり、動特性の考察が行なわれていない。その解析も1つの条件下における5種類の軸受の中の計算例の1つであり、スイングパッド軸受に関する考察は行なっていない。

また、動特性についてはバネ係数の主対角項が求めているだけで、安定性の考察がおこなわれていない。

以上述べてきたように、スイングパッド軸受の特性に関する研究は、ほとんどおこなわれていないと言える。

### 1-3 本研究の目的と方針

以上述べてきたように、スイングパッド軸受の解析は多くの問題があり、この軸受の特性は明かにされているとはいいがたい。本研究は、その点を考慮し、以下に述べる点に着目し、スイングパッド軸受の特性を明らかにして行く事を目的とする。

スイングパッド軸受を軸形に適用したときの安定限界特性の考察をする。

スイングパッド軸受を軸形に適用したときの不釣り合い応答を考察する。

また本研究の方針としては、スイングパッド軸受を支持軸受としたロータ系の不釣り合い応答と安定限界を計測するための実験装置を作る。計測項目は不釣り合い応答振幅と、安定限界速度とする。また、実験のパラメータとしては、スイングパッド軸受の軸受長さ（ $L/D$ ）、ロータの重量、偏重心させるための重量、軸受半径隙間（ $C/D$ ）とする。計測方法としては、軸系の振幅は渦電流式ピックアップで検出し、その信号を A/D 変換ボードを通しコンピュータに取り込みデータ処理する。

また、このデータより、不安定振動を起こし始めたときの回転数より安定限界速度を求める。軸受温度は、軸受の、潤滑油のインレットとアウトレットの温度の平均を取り求める。また回転数は電磁ピックアップで検出し、ユニバーサルカウンタで測定する。



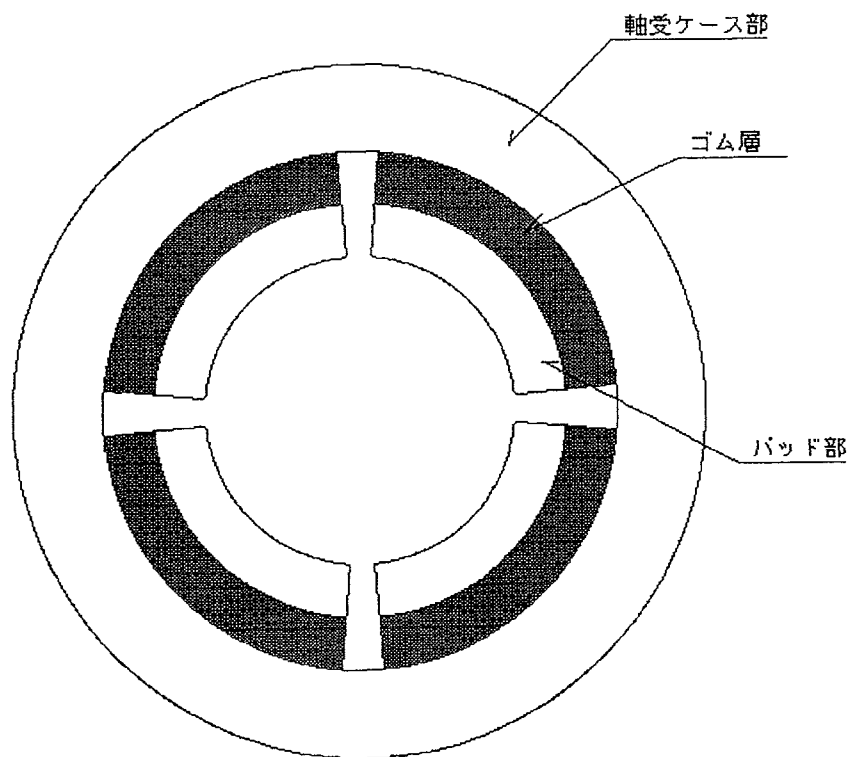


図 1 - 1 実験に使用したスイングパッド軸受

## 第2章 実験装置及び実験方法

### 2-1 実験装置の構成

本実験装置は、以下に述べるように構成されている。ロータ、軸受からなる実験装置本体。変速機付きモータ、プーリ、Vベルトからなる駆動系。ポンプ、バルブ、からなる給・排油系。ロータの変位、軸回転速度、潤滑油の供給温度を測るための測定系。そのデータを観測、記録する観測・記録系。図2-1にそのブロック図を示す。

## 2-2 実験装置概要

図2-2 に実験装置本体の正面図を示す。ロータは、17 の弾性軸とその中央に取り付けられた、18、19、20、のディスクからなる。

ロータの駆動は出力 3.7 K W の無段変速機付きモータより、2 段のプーリで増速する。モータの回転はインバータで制御する。またモータや、プーリ等の振動が測定に影響を及ぼさないように、駆動系と実験装置本体の軸とはゴムホースによって繋ぐ。

ロータを支持する 2 台の軸受台のうち駆動系側の軸受台には、A の単列深溝玉軸受が組み込む。もう一方の軸受台には、供試軸受ブロックが複列自動調心玉軸受を介して支持台に取り付ける。なお、駆動側の単列深溝軸受の剛性は、軸の曲げ剛性に比べて十分に大きい。また、供試軸受側の自動調心軸受は回転させず、回転軸の心を出すためである。

ロータの両側に設けられた 1 と 3 の振れ止めは、不慮の事故を想定し、振幅が一定限界値を越えた時に、内側に取り付けられた無給油摺動材と軸が接触し、振幅を制限するためである。

給・排油系のブロック図を図2-3 に示す。潤滑油はオイルタンクより給油ポンプで加圧され、ストレーナとオイルクーラを通り給油ホースより軸受ブロックの入口より供試軸受に供給される。そこで潤滑油は両端の軸受の隙間を軸方向に流れる。そして、潤滑油はオイルパンに集められ、排油ポンプにより再びオイルタンクに戻される。オイルタンクではオイルクーラにより熱交換される。これらのオイルクーラは、潤滑油の温度を一定に保つために設けた。

回転速度は、駆動系の増速された方の軸に歯車を取り付け、そこにセンサを近づけて信号を読みとり、測定する。軸の X-Y の変位はディスク部分の変位を図2-4 のようにセンサを取付て測定する。この信号を FFT アナライザとコンピュータに取り込み、測定する。

次に以下に各部詳細を述べる。

### 2-2-1 供試軸受

本実験ではスイングパッド軸受を単純モデル化するため、また、加工を容易にするためゴム層とパッド層を1層ずつとした。本実験に使用したスイングパッド軸受と真円軸受を図2-5 a, 2-5 b にそれらの写真を写真1, 2 に示し、各部詳細を表 2-1 に示す。

本実験ではスイングパッド軸受の長さを2種類用意した。また、スイングパッド軸受と真円軸受の比較をするため、真円軸受も1種類用意した。スイングパッド軸受と真円軸受の比較において、同一条件にするため両者ともジャーナルに当たる部分のパッド及び軸受面はブロンズ（BS）とした。

### 2-2-2 軸

図2-6 軸の作製図（軸 1）を、その写真を写真3 に、そして表2-2 に軸の諸元を示す。軸は軸受半径隙間を変えるためにジャーナル部の直径が違うものを3種類用意した。また軸は、実用的かつ駆動系からの安定した回転速度をえられる状態での運転域で結果を得るために、以下のよう設計した。

駆動系から得られる回転速度以内で危険振動回転速度を得るために、軸の固有振動数を低くする。これを実現するため、軸の主要部をなるべく細くしかつ長く設計する。

軸自体の不釣り合いをなるべく小さくするため、軸直径を小さくする。左右の軸受に当分に負荷がかかるように、軸の左右の重量は同じにする。

### 2-2-3 ディスク

図2-7 にディスク（Disk I）の製作図を、表2-3 にディスクの諸元を示す。各ディスクにおいて、製作上、材質上の不釣り合いが大きいと、実験を行う際に不具合を生じる。これを除去するため、ディスクにあらかじめ設けた、不釣り合いを取るための穴に、セットスクリュー（2種類の長さのM4のネジと座金を使用）を取り付ける。これにより危険速度通過時の振幅を抑える。

#### 2-2-4 軸受ブロック

図2-8 に供試軸受ブロックの製作図を示す。 供試軸受ブロックの材料は SS41 である。 供試軸受を軸受ブロックの間に夾む。 潤滑油は軸受ブロックの外側上下部から内側へと供給する。  
また、軸受ブロックには温度測定用の熱電対を取り付けるため穴が 2 箇所設けている。 この熱電対により潤滑油の温度を測定する。

#### 2-2-5 駆動系

図2-9 に駆動系の概略図を示す。 本実験では、モータと実験装置本体の間にプーリと V ベルトを用いた 2 段の増速機により、実験装置本体の軸回転速度を 3000 rpm から 8000 rpm まで増減速が可能である。  
表2-4 に駆動系の仕様を示す。( ) 内の数字は図2-9 参照

#### 2-2-6 給・排油系

図2-3 に給・排油系のブロック図を示す。 潤滑油は、タンクに貯えられており、そこより給油ポンプにより加圧され、ストレーナを通る。  
この時のストレーナは潤滑油に混入した異物を取り除く為に設けられている。 ここを通った潤滑油はオイルクーラに行く。 オイルクーラは、潤滑油の温度を調節するために設けた。 そして、この潤滑油は軸受ブロックに入り供試軸受に供給される。 その後、潤滑油は圧力がかからないように開放状態になった穴よりオイルパンに落ちる。 オイルパンに落ちた潤滑油は、排油ポンプに行き加圧され再びオイルタンクに戻る。 上に記したオイルクーラのみでは潤滑油の温度調節はできないためオイルタンクにもオイルクーラを取り付ける。

#### 2-2-7 測定系

測定系のブロック図を図2-10、仕様は表2-5に示す。 軸の回転速度は駆動系の増速された最終の軸に歯車を取り付け、電磁式回転検出器をその歯車に接触しないように取り付ける。 そのピックアップをユニバーサルカウンタに接続する。 この時歯車の歯は 60 枚なので、ユニバー

サルカウンタの読みが直接 rpm になる。 回転速度は、このカウンタの数値を読み測定する。

潤滑油の温度測定は銅-コンスタンタンの熱電対を用いる。 軸受ブロックには熱電対を取り付けるための穴が設けられており、そこに熱電対を取り付ける。 熱電対は導線を通しポータブルハイブリットレコーダに取り付ける。 このハイブリットレコーダには自動的に温度校正をする装置が内蔵されているため、冷接点は必要としない。 潤滑油の温度は、このレコーダより直接読みとり測定を行う。 本実験に用いた潤滑油は、温度による粘土の変化は比較的小さいのであるが、粘度を一定に保つため、設定温度を  $\pm 5^{\circ}\text{C}$  とした。この温度範囲であれば、 $35^{\circ}\text{C}$  を基準にした粘度 ( $\mu = 8.8$ ) の変化の割合は  $\pm 2.2\%$  以内となる。

軸の x-y 方向の変位の測定は、水平、鉛直方向に、それぞれに 1 つずつの非接触変位計を取り付ける。 この変位計の信号を変換器に取り入れ、増幅する。 変換器で増幅された信号は AD 変換ボードに取り入れられ、デジタル信号に変換されてからパーソナルコンピュータに取り込む。 この時、パーソナルコンピュータでリアルタイムに処理していると遅くなるため、一旦フロッピーディスクに貯え、後からこのデータを処理することとする。 また変換器より得た信号は並列に FFT アナライザにつながっており、リアルタイムに処理された振動周波数波形が測定できる。 この波形を観察することにより安定限界速度の測定を行う。 また、動特性を調べる上で、軸自体の X-Y 方向の変位を調べるより、ディスクの X-Y 方向の変位を調べるのが望ましいため、図 2-4 の様にディスクの水平・鉛直方向に非接触変位計のプロブを取り付け、ディスクの振動を測定する。 水平方向を検出するプロブは、軸支持台上にマグネットスタンドを用いて固定し、鉛直方向。のプロブは山型鋼を加工して固定する。

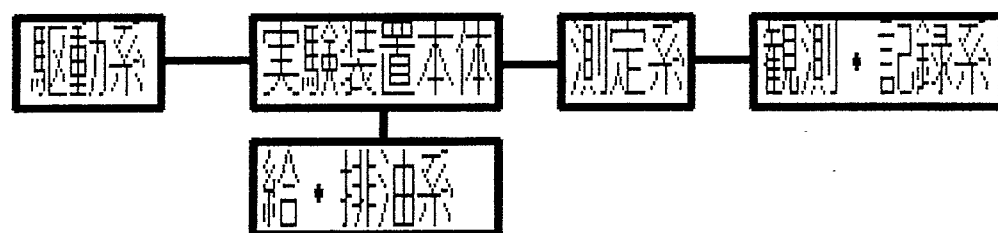


図 2 - 1 実験装置全体のブロック図

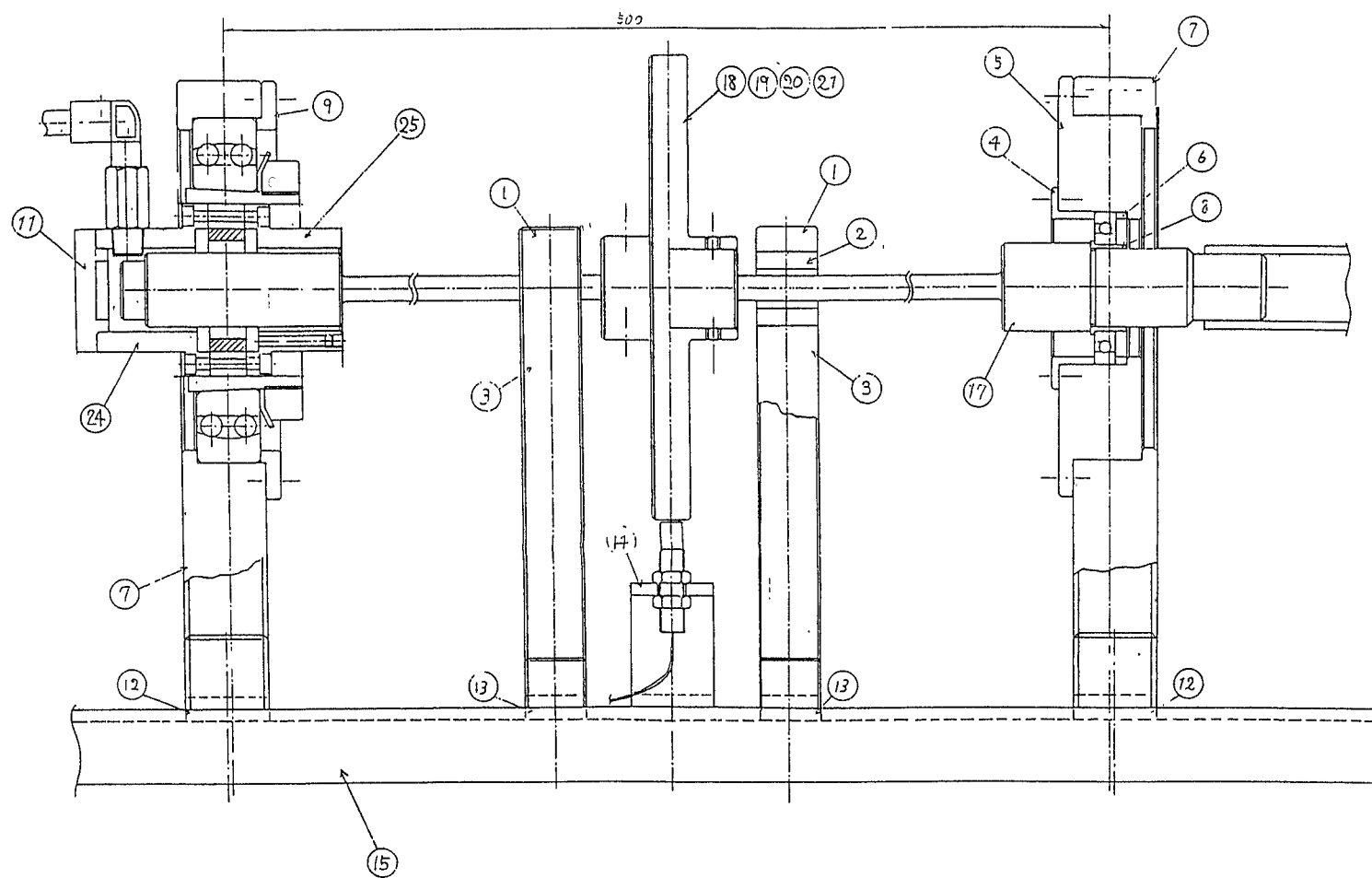


図 2 - 2 実験装置本体の正面図



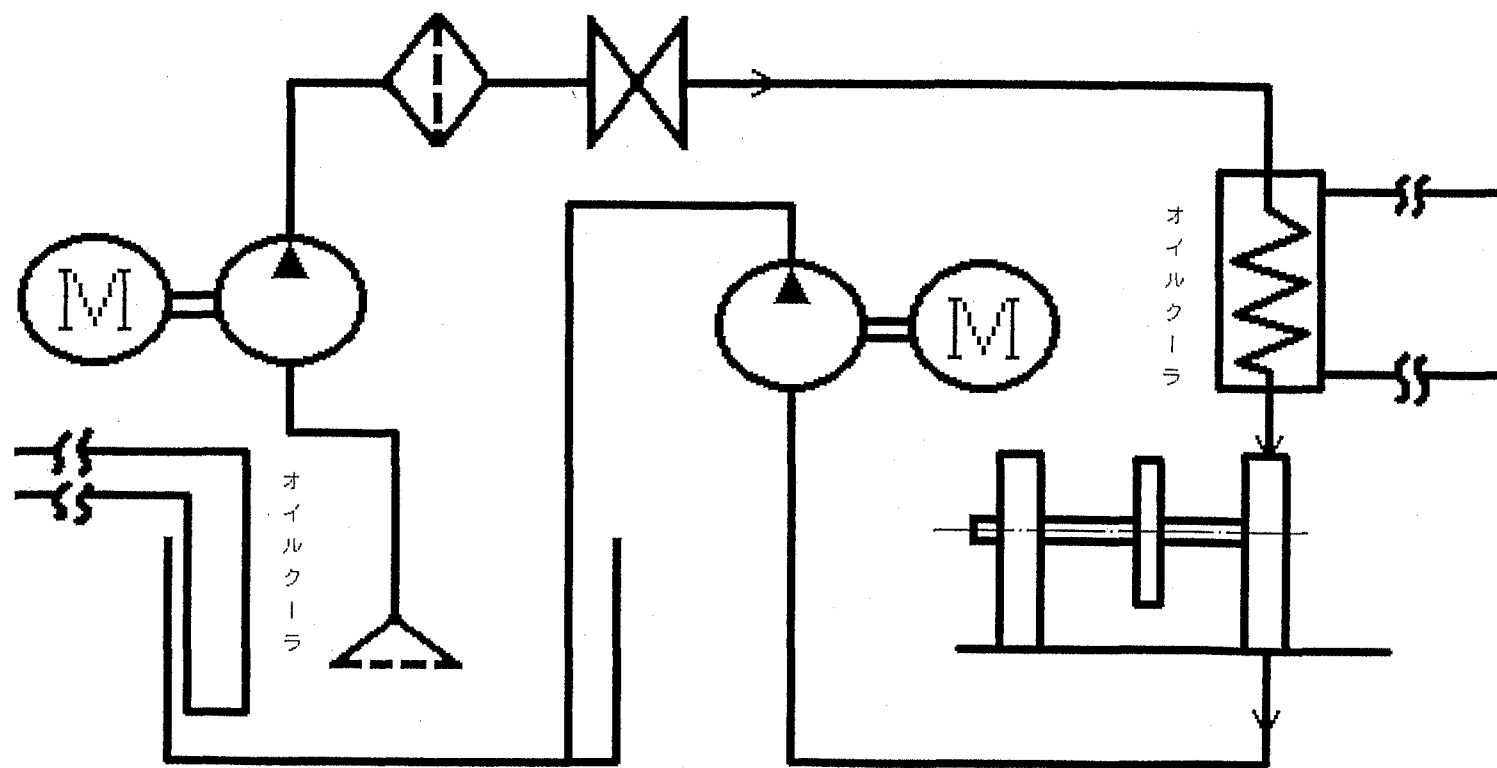


図 2 - 3 給・排油系のブロック図

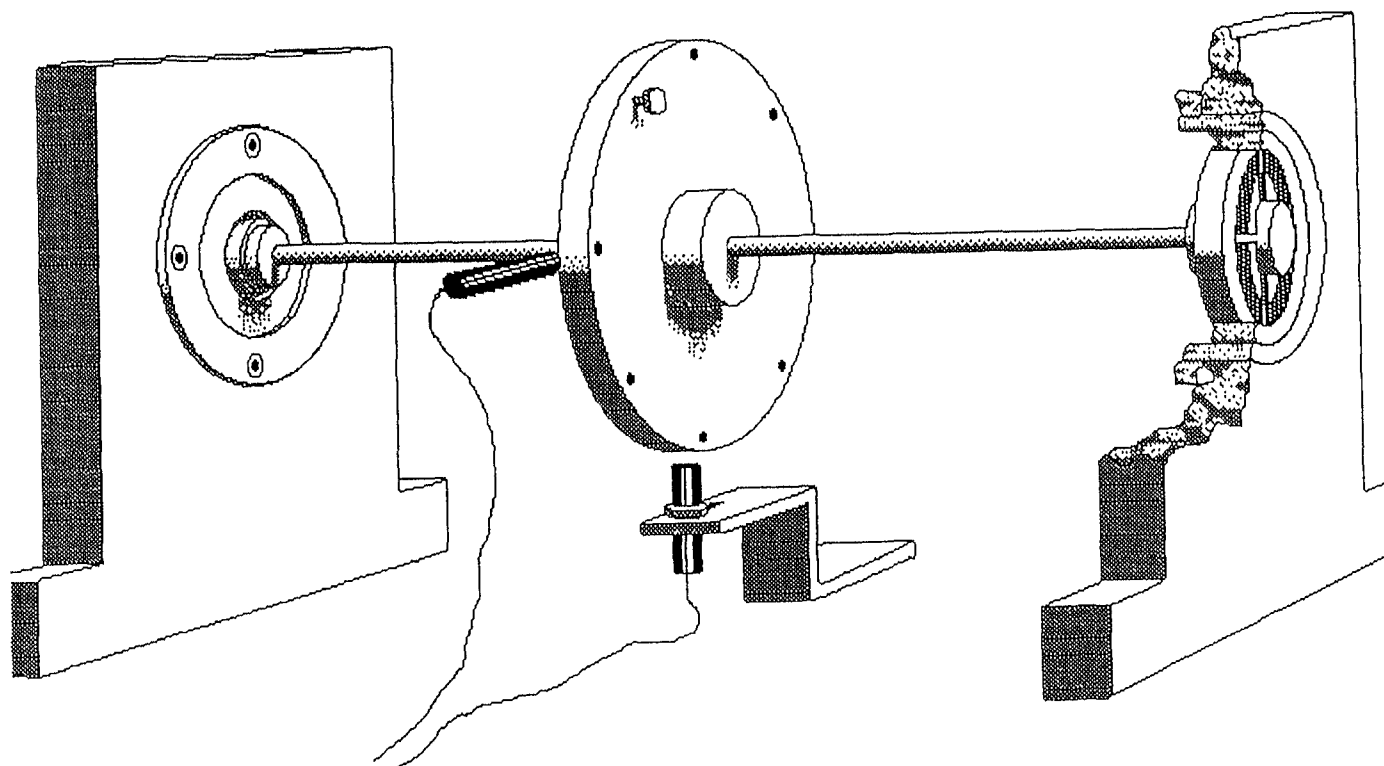


図 2 - 4 非接触変位計のプロープの取り付け位置

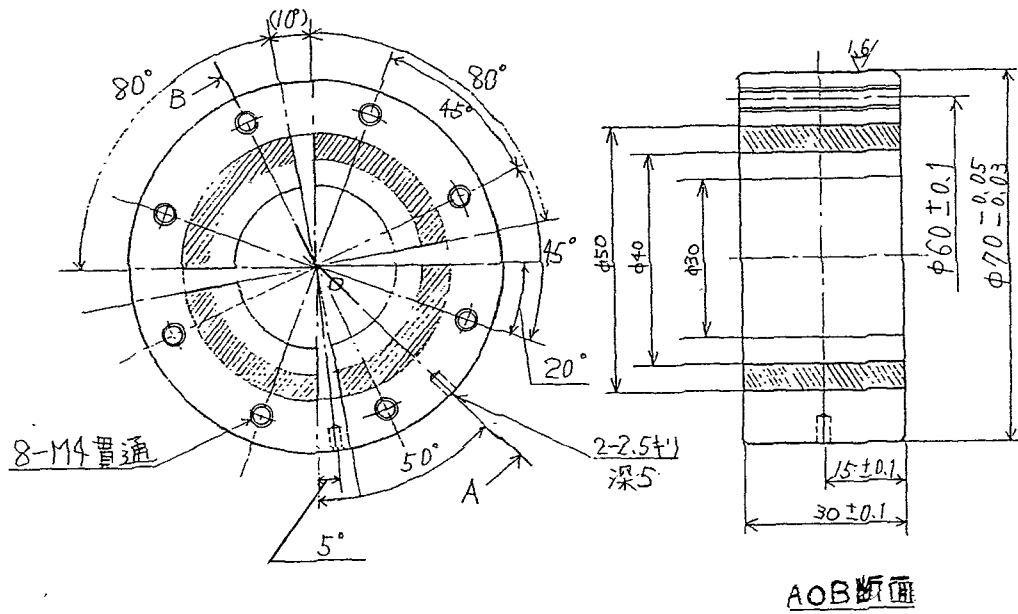


図 2 - 5 a スイングパッド軸受の作製図

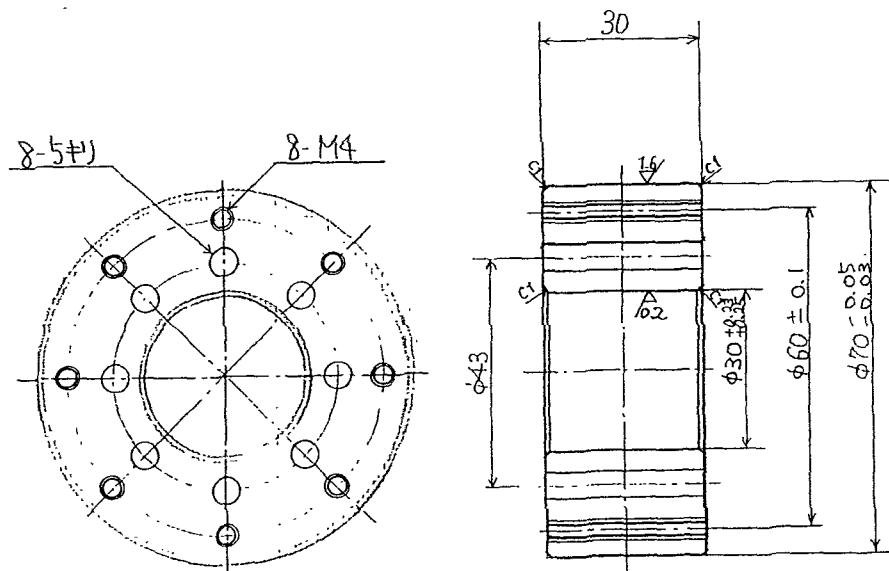


図 2 - 5 b 真円軸受の作製図

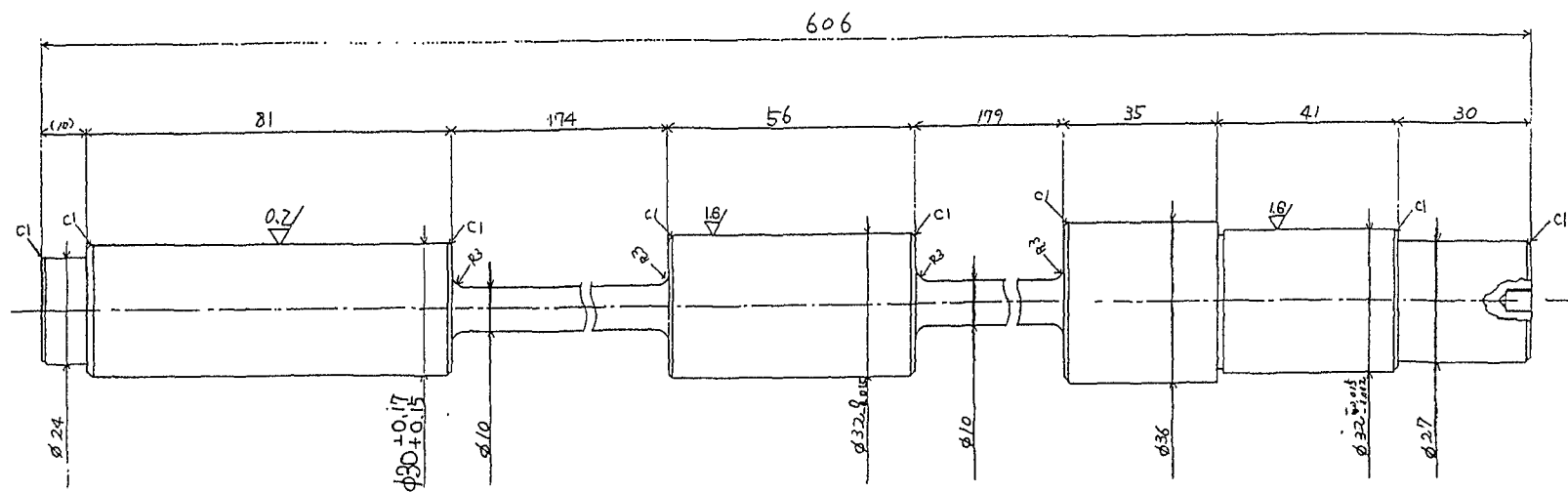


図 2 - 6 軸の作製図

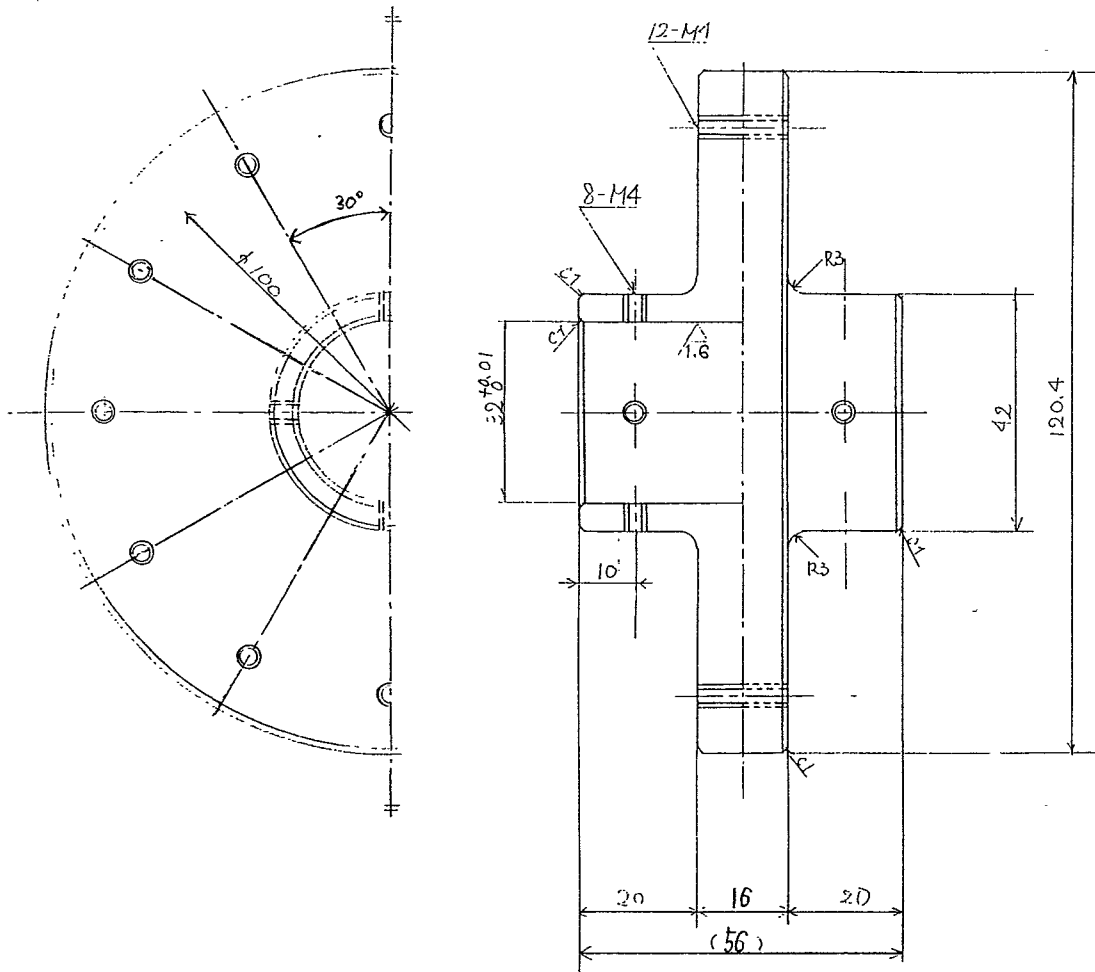


図 2-7 ディスクの作製図

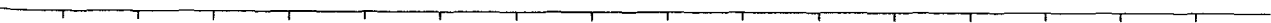


図 2-8 軸受ブロックの作製図

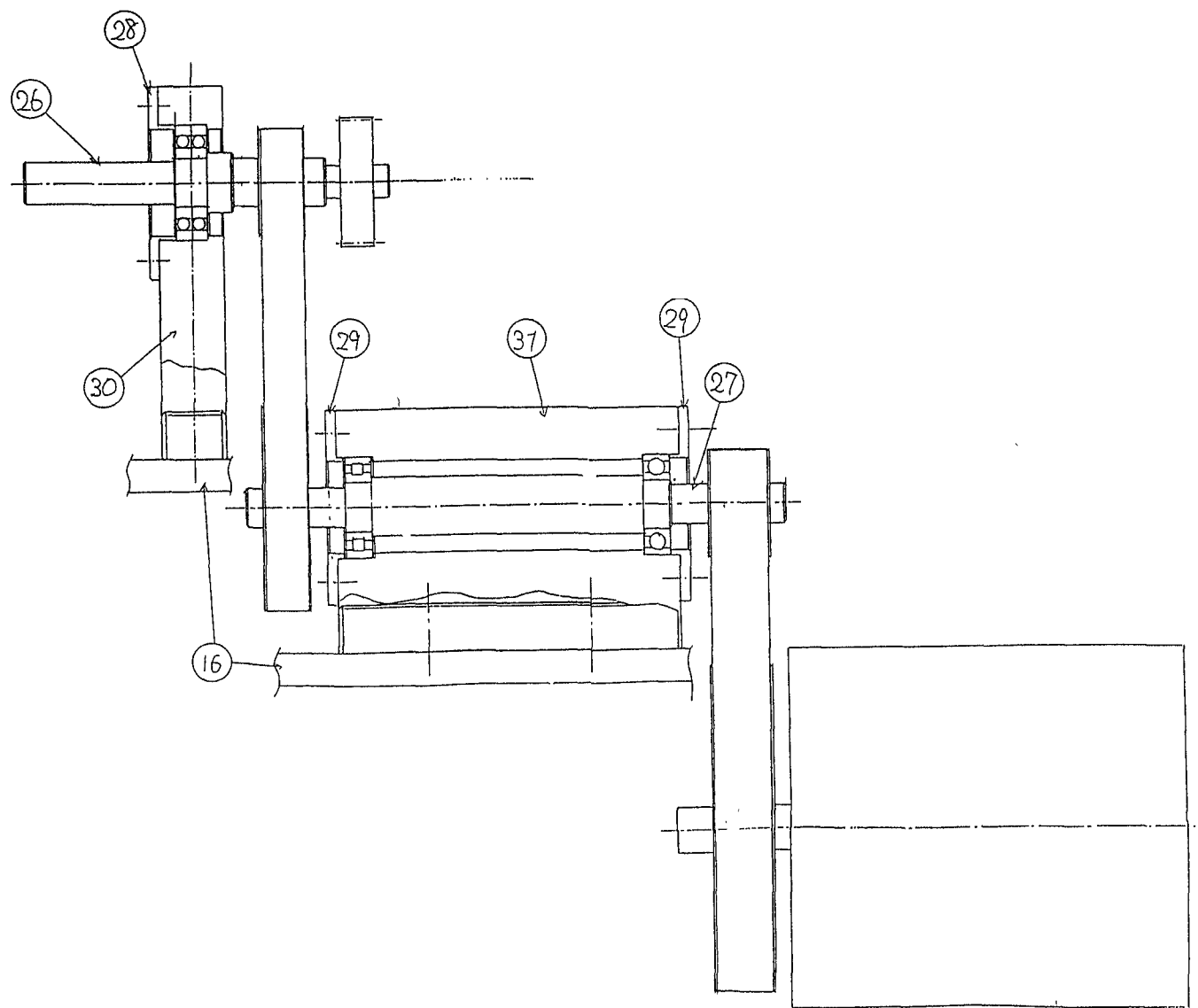


図 2 - 9 駆動系の概略図

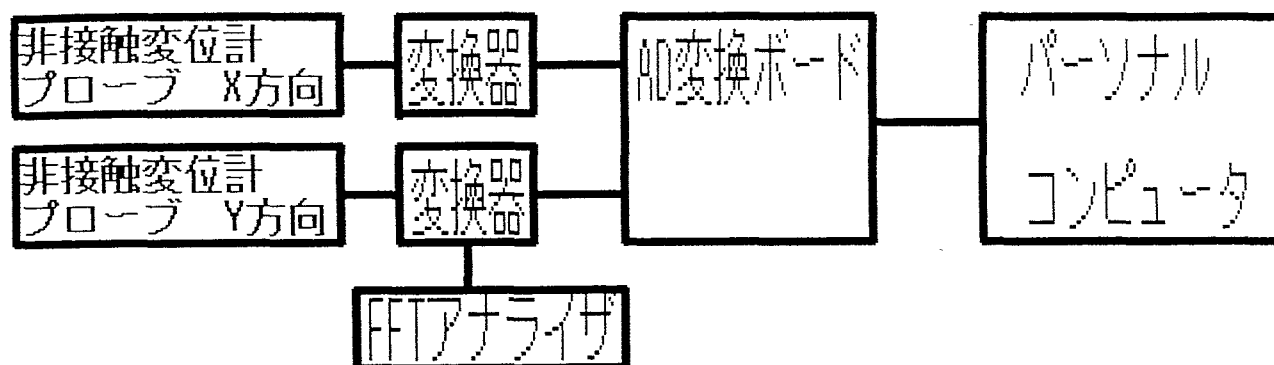


図 2 - 1 0 測定系のブロック図



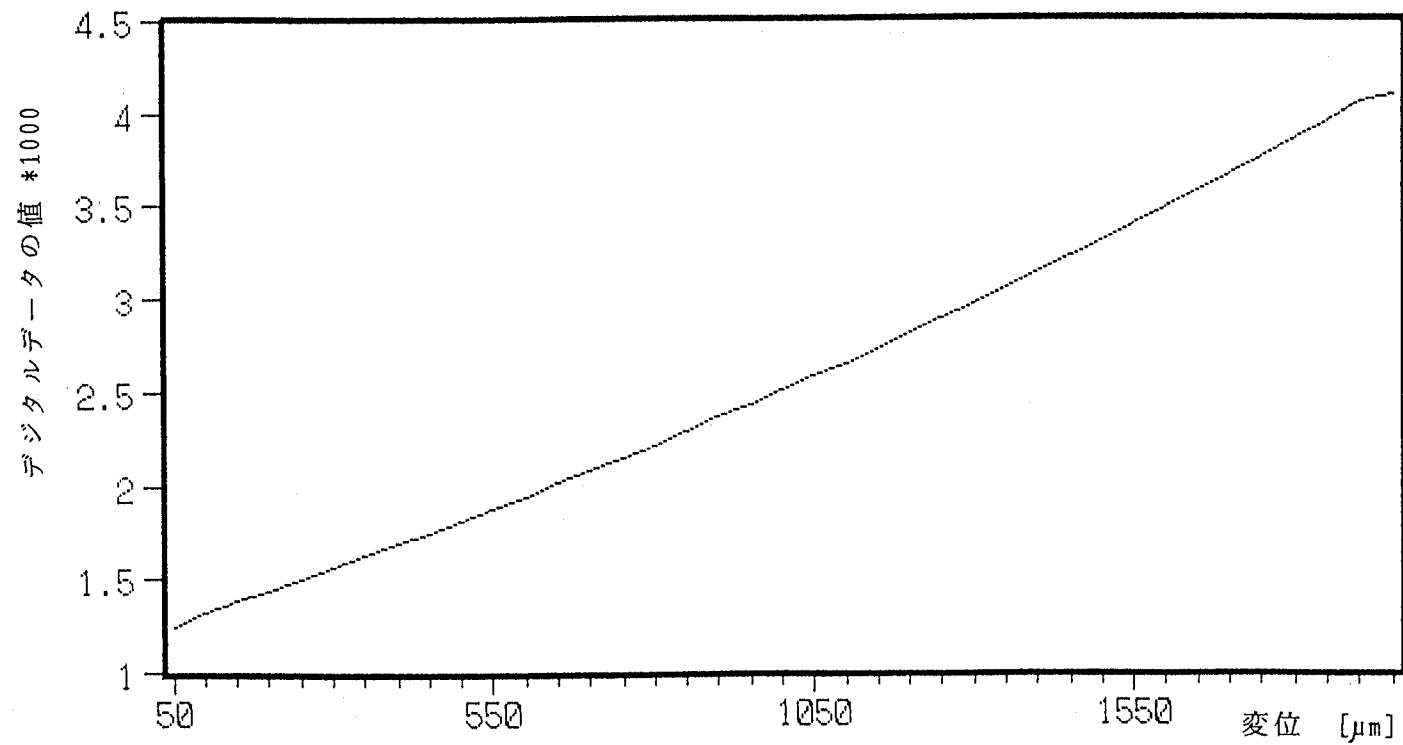


図 2 - 1 1 a 水平方向に取り付けられた  
非接触変位計の校正

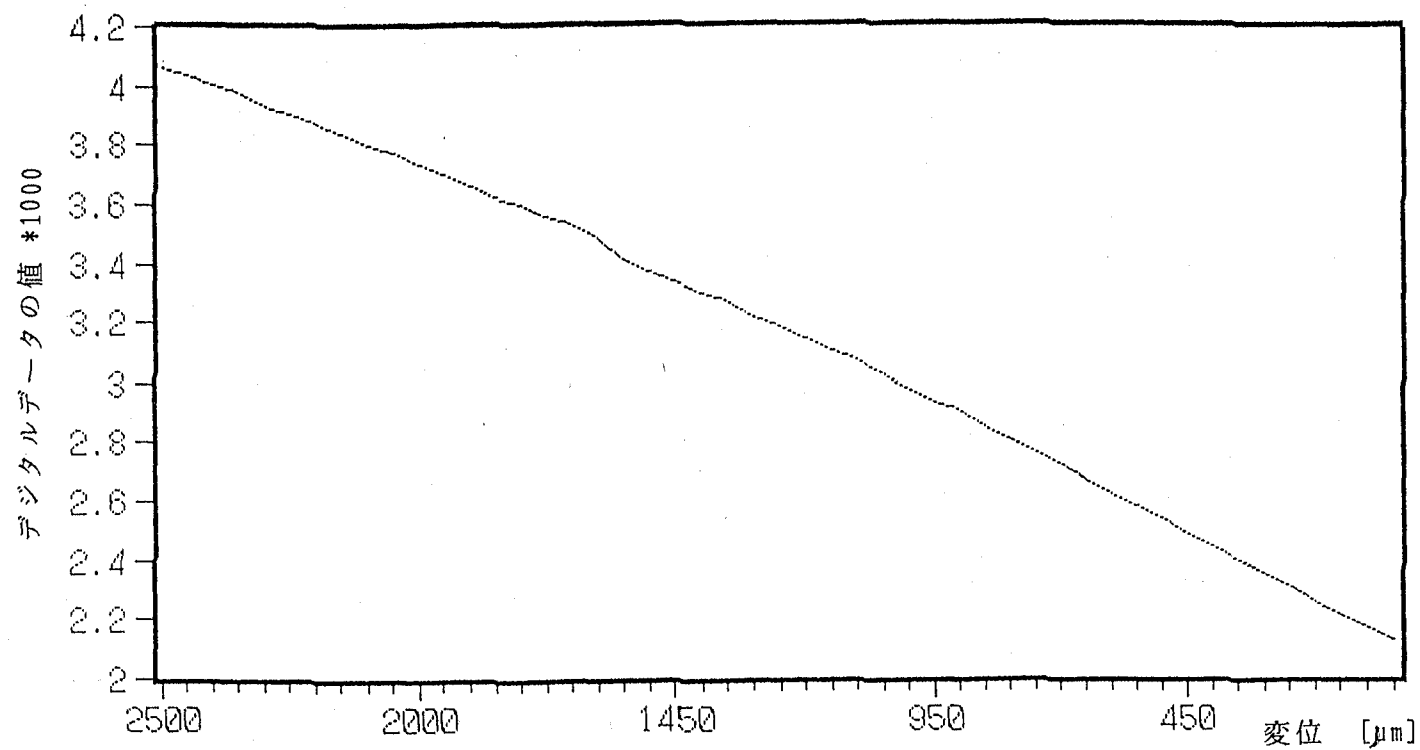


図 2 - 1 1 b 鉛直方向に取り付けられた  
非接触変位計の校正

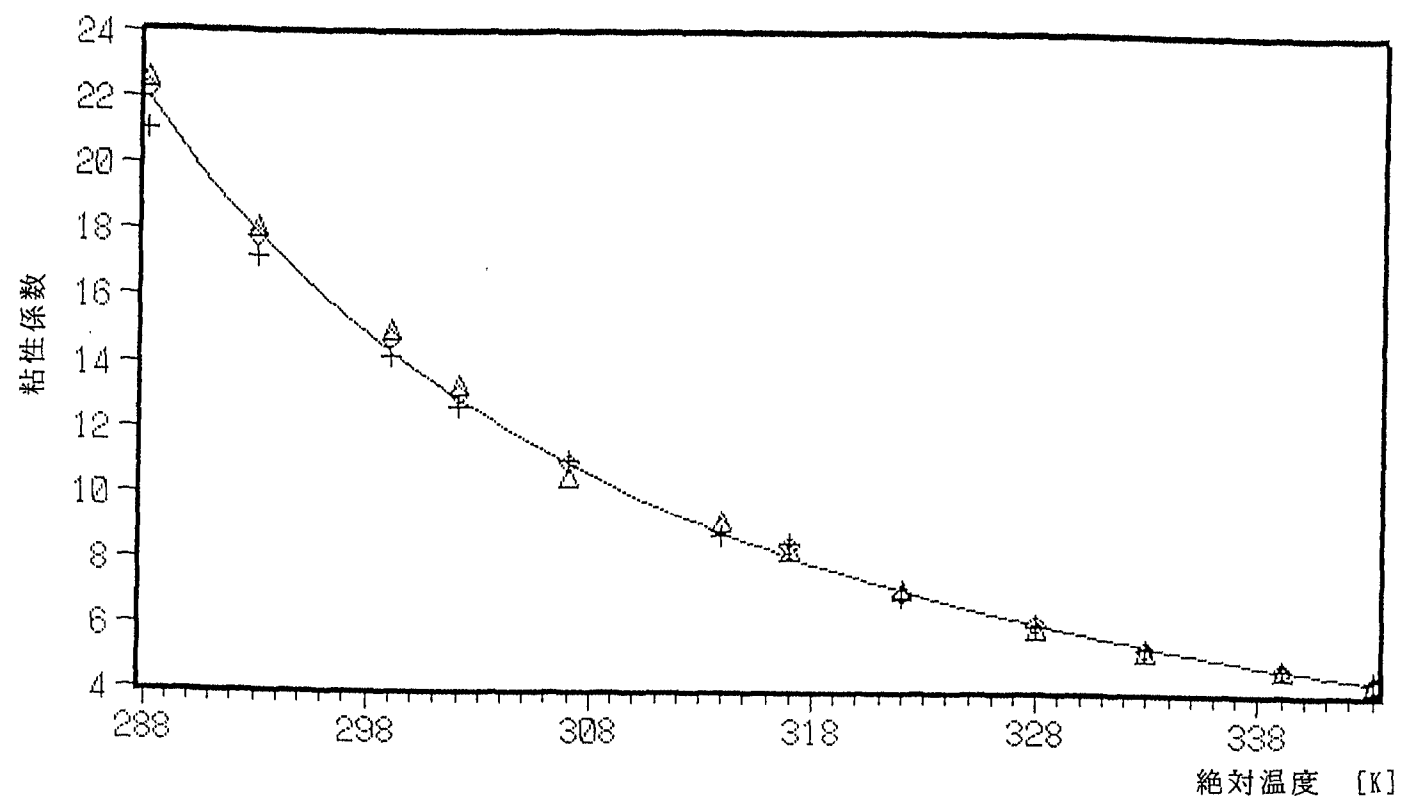


図 2 - 1 2 潤滑油の粘度と温度の関係



写真1 実物のスイングパッド軸受の写真



写真2 実物の真円軸受の写真

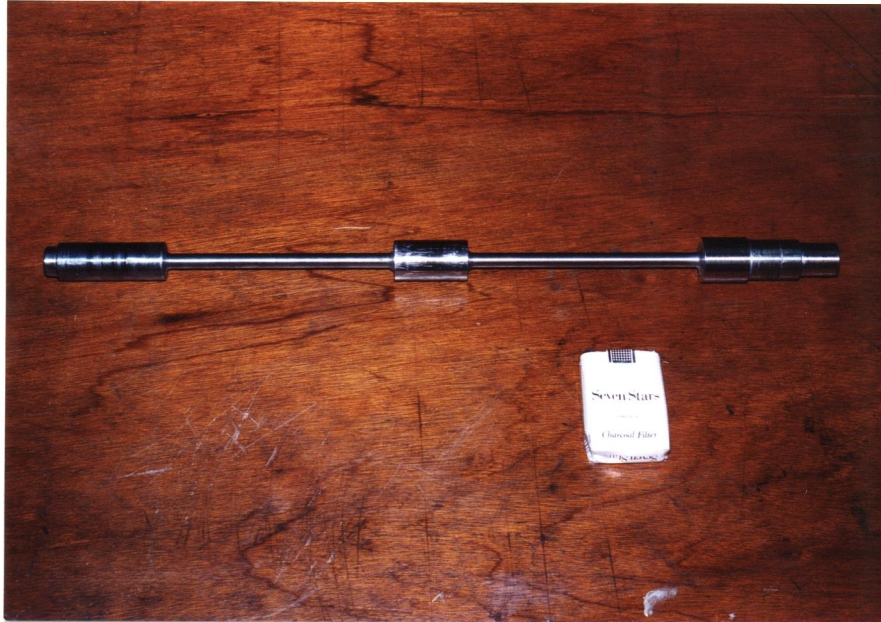


写真3 実物の軸の写真

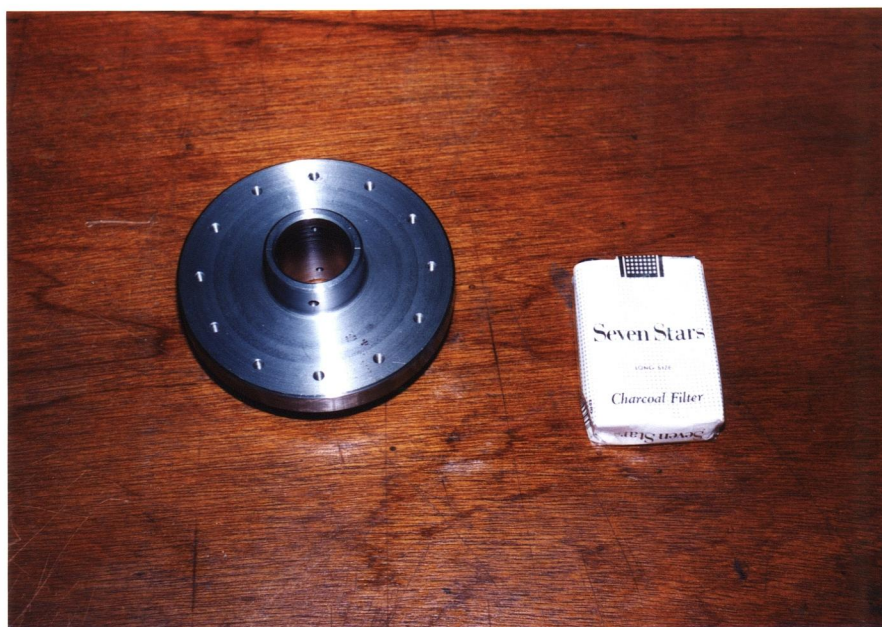


写真4 実物のディスクの写真



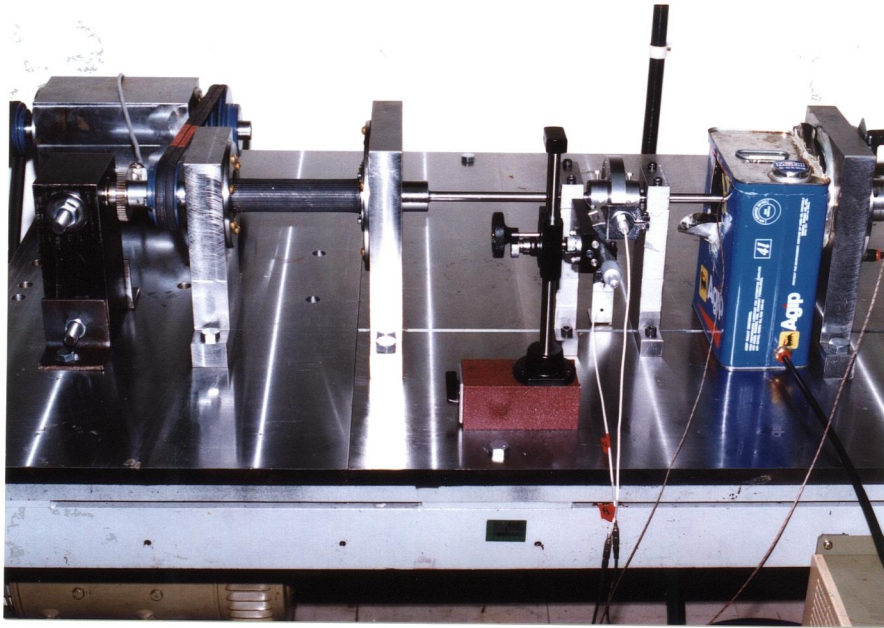


写真5 実験装置本体の写真

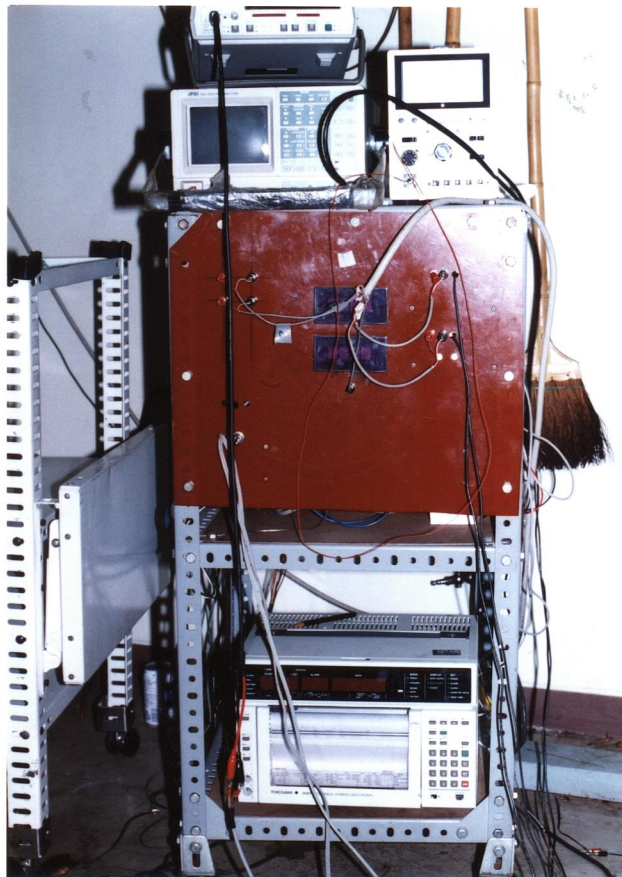


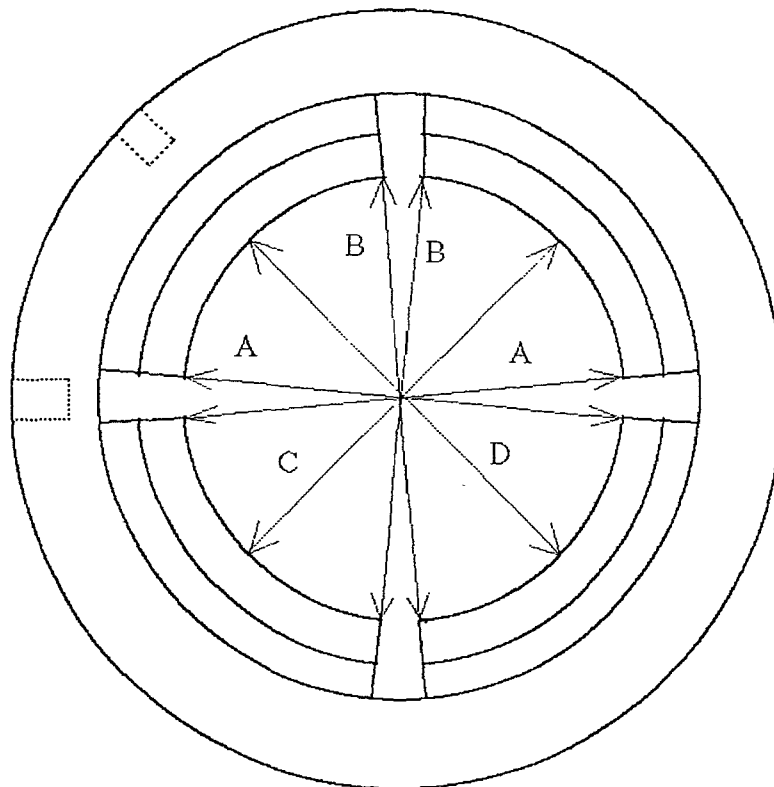
写真6 測定装置の写真

スイングパッド軸受径		平均
L = 3 0		
A	30.26	30.24
A'	30.23	
B	30.30	
B'	30.23	
C	30.22	
D	30.20	
L = 2 2 . 5		
A	30.22	30.22
A'	30.23	
B	30.21	
B'	30.24	
C	30.2	
D	30.19	

真円軸受 (mm)		平 均
L = 3 0		30.2
A	30.21	
B	30.20	
C	30.20	
D	30.20	
材 質		B S

ゴム 材質	N B R	H s = 7 0 °
貯蔵弾性係数		1 5 Kg/cm <sup>2</sup>
測定 条件	2 0 ° C	圧縮方向
パッド部材質		B S

表 2 - 1 スイングパッド軸受と真円軸受の各部詳細





軸 1 $\phi 10$	
軸受部直径 (mm)	
外側垂直方向	30.00
外側水平方向	30.00
中央垂直方向	30.00
中央水平方向	30.00
内側垂直方向	30.00
内側水平方向	30.00
平 均	30.00
重 さ (g)	1721.5
長 さ (mm)	600
材 質	S 4 5 C

軸 2 $\phi 12$	
軸受部直径 (mm)	
外側垂直方向	30.07
外側水平方向	30.06
中央垂直方向	30.06
中央水平方向	30.05
内側垂直方向	30.07
内側水平方向	30.06
平 均	30.06
重 さ (g)	1829.0
長 さ (mm)	600
材 質	S 4 5 C

軸 3 $\phi 12$	
軸受部直径 (mm)	
外側垂直方向	29.95
外側水平方向	29.95
中央垂直方向	29.95
中央水平方向	29.95
内側垂直方向	29.95
内側水平方向	29.95
平 均	29.95
重 さ (g)	1800.0
長 さ (mm)	600
材 質	S 4 5 C

リング部	
重 さ (g)	34.6
材 質	S 4 5 C

表 2 - 2 各軸のジャーナル部の寸法

D i s k	1	2	3
径 (mm)	82.55	95.05	107.30
重 さ (g)	731.5	945.5	1492.0
表面誤差	$\pm 20 \mu m$	$\pm 40 \mu m$	$\pm 45 \mu m$
材 質	S 4 5 C		

重りの平均の重さ (g)	
ボルト (長)	1.740
ボルト (短)	1.541
座 金	0.289

表 2 - 3 ディスクの各部詳細

表 2 - 4 駆動系各部の仕様

原動機 (10)

MIKI PULLEY (株) 製	
MODEL	SM-3700
出力	3.7KW
ARM VOLT	160V
E I E L D V O L T	185V
BASE RPM	1750 r p m
ARM CUR	26.2A

増速機 1 段目 (増速比 2.67)

Vベルト (15)	3V-630
駆動側プーリー (14)	呼び径 200 < 3V-200 >
従動側プーリー (13)	呼び径 75 < 3V-75 >

増速比 2 段目 (増速比 1.67)

Vベルト (16)	3V-560
駆動側プーリー (12)	呼び径 125 < 3V-125 >
従動側プーリー (6)	呼び径 75 < 3V-75 >

軸受台 (17)

軸受	深溝玉軸受	< 6206 >
軸受	円筒ころ軸受	< NU 206 >

軸受台 (11)

軸受	アンギュラ玉軸受	< 7206C >
----	----------	-----------

軸受台 (19)

軸受	深溝玉軸受	< 6805 >
----	-------	----------

表 2 - 5 測定系の各部の仕様

非接触変位計 (株) 電子応用  
 変換器 GAP-SENSOR ACE-5509  
 プローブ PU-09  
 ケーブル PC-03

AD変換ボード (株) タートル工業  
 T98-0812ADT

パーソナルコンピュータ NEC  
 PC-9801RX2

ファンクションジェネレータ キクスイ電子(株)  
 RC OSCILLATOR model 417A

FFTアナライザ AND Co, ltd  
 AD-3522

振動計 リオン株式会社  
 VM-20A

ポータブルハイブリットレコーダ YOKOGAWA  
 3087

電磁式回転検出器 (株) 小野測器  
 MP-911

ユニバーサルカウンタ (株) 小野測機  
 MT-3100

潤滑油 Mobil 石油  
 Velocite oil No, 6 (VG10)

回帰分析の結果：	
Y 切片	2151.8930
Y 評価値の標準誤差	35.236721
R 2 乗	0.9964392
標本数	49
自由度	47
X 係数	0.7936965
X 係数の標準誤差	0.0069206

表 2 - 6 a 水平方向に取りつけられた  
非接触変位計の校正結果

回帰分析の結果：	
Y 切片	1071.9622
Y 評価値の標準誤差	61.413925
R 2 乗	0.9950461
標本数	39
自由度	37
X 係数	1.5065506
X 係数の標準誤差	0.0174756

表 2 - 6 b 鉛直方向に取りつけられた  
非接触変位計の校正結果

軸の固有振動数 (H z)			
	D i s k 1	D i s k 2	D i s k 3
軸 1	48.0	46.5	45.0
軸 2	69.0	67.0	65.0
軸 3	68.0	66.0	64.0

表 2 - 7 各軸の固有振動数

チャンネル 1	温度計との差	チャンネル 2	温度計との差	油温度	水の温度
80.6	2.6	80.5	2.5	78	80
69.6	2.8	69.2	2.4	66.8	66.8
62.4	2.6	62	2.2	59.8	59.9
54	3	53.5	2.5	51	50.5
48.7	2.7	48.5	2.5	46	45.3
43.4	2.9	43.2	2.7	40.5	40
39.1	3.1	38.8	2.8	36	36
33.6	3.5	33.3	3.2	30.1	30
28.8	2.8	28.8	2.8	26	25.5
23.8	2.8	23.8	2.8	21	20.3
18.7	3.2	18.6	3.1	15.5	16.2
平均温度差	2.999	平均温度差	2.6818		

表 2 - 8 熱電対の校正

温度	時間	比重	動粘度	粘度
343	78.01	0.811	4.454371	3.6124948
343	76.02	0.811	4.340742	3.5203417
343	76.53	0.811	4.369863	3.5439588
339	82.7	0.813	4.72217	3.8391242
339	83.52	0.813	4.768992	3.8771904
339	82.99	0.813	4.738729	3.8525866
333	93.69	0.816	5.349699	4.3653543
333	93.87	0.816	5.359977	4.3737412
333	92.61	0.816	5.288031	4.3150332
328	104.59	0.819	5.972089	4.8911408
328	105.77	0.819	6.039467	4.9463234
328	103.33	0.819	5.900143	4.8322171
322	120.83	0.825	6.899393	5.6919992
322	121.54	0.825	6.939934	5.7254455
322	123.72	0.825	7.064412	5.8281399
317	148.62	0.829	8.486202	7.0350614
317	147.39	0.829	8.415969	6.9768383
317	145.65	0.829	8.316615	6.8944738
314	152.48	0.831	8.706608	7.2351912
314	157.18	0.831	8.974978	7.4582067
314	160.26	0.831	9.150846	7.6043530
306.5	192.89	0.835	11.014019	9.1967058
306.5	190.94	0.835	10.902674	9.1037327
306.5	184.33	0.835	10.525243	8.7885779
302	220.63	0.838	12.597973	10.557101
302	227.9	0.838	13.01309	10.904969
302	232.53	0.838	13.277463	11.126513
299	248.1	0.84	14.16651	11.899868
299	257.15	0.84	14.683265	12.333942
299	262.22	0.84	14.972762	12.577120
293	301	0.843	17.1871	14.488725
293	311.96	0.843	17.812916	15.016288
293	316.93	0.843	18.096703	15.255520
288	368.57	0.846	21.045347	17.804363
288	388.73	0.846	22.196483	18.778224
288	396.03	0.846	22.613313	19.130862

表 2 - 9 潤滑油と温度と比重の関係

## 2-3 実験方法

### 2-3-1 非接触変位計の校正

(1) 正しい非接触変位計の校正を行う為には実際行う実験の装置を組み上げ、その装置を用いて校正を行わなければならない。今回の非接触変位計の校正は、実際に軸にディスクを取り付けた状態で行う。

次に、非接触変位計のプローブを非接触変位計の校正用のマグネットスタンドに取り付ける。プローブは軸方向と垂直に成っていることを確かめる。(プローブの先の面を軽くディスクに付けてみて隙間ができていないか見てみる。)

(2) 変換器、パーソナルコンピュータの電源をいれ、コンピュータを測定できるよう準備をする。準備ができれば、適当にプローブの高さと距離を決める。次に、適当に高さを動かしながら非接触変位計から出力された電圧値を読む。この時一番小さい電圧値になった時が最適な高さとなる。この位置はプローブが軸に水平になった位置である。位置が決まると、次はプローブをディスクに付かない程度まで近づける。

(3) 非接触変位計を等間隔づつはなす。その時の電圧値と距離を記録する。今回は  $50\mu\text{m}$  づつ 50 箇所を測定した。この時、注意すべきことは動かしすぎても戻さないようにする。

(4) 本実験では X-Y 方向に 2 カ所に非接触変位計を使用するため、(1)～(4)の手順をもう一方のチャンネルにおいても同様に行う。これらの結果を図2-11 a, b に示す。(この時の値は付録に記す。) この結果より電圧と距離の関係を求め、その式を本実験のロータ変位の測定に使用した。

### 2-3-2 軸の固有振動数測定

測定したい軸の両端を自由端として固定する。軸に圧電式加速度型振動ピックアップを軸に両面テープで取り付ける。この振動ピックアップを振動計に接続し振動計の出力を FFT アナライザに取り付ける。



軸をハンマで軽く叩くことによりホワイトノイズを発生させる。加振された軸は固有振動数で共振するためFFTアナライザの波形出力には固有振動数近辺で大きくなる。この時のピークを示す周波数が軸の固有振動数である。以上により求めた軸の固有振動数の結果を表2-7に示す。

### 2-3-3 熱電対の温度校正

前項で書いたようにポータブルハイブリッドレコーダは内部に校正回路が内蔵されているため、冷接点は必要としない。しかしながら、正しい温度を得るために温度校正を行う。その温度校正は、以下のように行う。水槽にお湯を入れ、その中に潤滑油の入ったメスシリンダを入れる。お湯と潤滑油の中にアルコール棒温度計を入れる。お湯と潤滑油が目的の温度になり、かつそれぞれの温度差が小さくなった時にハイブリッドレコーダの値を読む。この動作を数回繰り返す。この時の熱電対と、それぞれの温度計の読みの値を表2-8に示す。

### 2-3-4 潤滑油の粘度と比重の測定

水槽にお湯を満たし、潤滑油を入れたメスシリンダをその中に入れる。そのメスシリンダの中に、ウベローテ型粘度計、アルコール棒温度計、比重計を入れる。前項同様に水槽のお湯を目的の温度に近づけ、お湯と潤滑油の温度差が小さくなった時、その時の温度、比重、粘度を求める。比重は比重計の値を読み記録しておく。粘度の測定は以下のように行う。ウベローテ型粘度計は一定量の潤滑油を高い位置に引き上げ細い管を降りてくる時間経過を測定する。この値に定数を掛けると動粘度が求まる。今回用いたウベローテ型粘度計の定数は0.00571である。この求めた動粘度に比重を掛けることにより潤滑油の粘度を求めることができる。潤滑油の粘度の温度のデータを用い最小二乗法より、潤滑油の粘度と温度の関係式を求める。この式を用いて実験の潤滑油の粘度を求める。この時の、温度、比重、動粘度、粘度、最小二乗法による式の値を表2-9と図2-12に示す。

### 2-3-5 ロータ系の不つり合い除去

ロータ系はディスクと軸により構成されているが、ディスクの製作上、材質上の不つり合いが実験に大きな影響を及ぼすと思われる。

これを取り除くため、ディスクにあらかじめ設けたバランス取り用の穴にセットスクリュー（M4：各種の長さを用意）を取り付けることによって、危険速度通過時の振幅を抑えることにする。

以下にその手順を述べる。

（１）ディスクはセットスクリュー8本で軸に取り付けられているが、この8本のセットスクリューの締め付け具合を調節し、ディスク表面に指針を当てたダイヤルゲージを用いて静的変位を数十 $\mu\text{m}$ におさえる。

（２）図2-7に示すディスク側面のつりあいおもり穴のうち、90度おきに任意の4箇所を選び、それぞれの位置についてつりあいおもりを取り付け、応答曲線を描かせ振幅の小さい2箇所を選定する。

（３）得られた2箇所の間の2つの穴についても、同様な作業を繰り返す、危険速度通過時の振幅が最小になる所を見いだす。

（４）1箇所のバランスおもりで不足な場合は、その周囲の穴にもおもりを加え、最良の状態が見つかるまで、（２）と（３）の作業を繰り返す。

本来ならば、危険速度通過時の振幅をゼロにおさえるべきものなのであるが、本実験では上記の方法を用いて得られた最適の条件、すなわち、危険速度におけるロータの振幅が最小になる条件を採用する。

### 2-3-6 測定、観測・記録方法

本実験では、次の項目を測定する。

（１）ロータの振幅（水平・鉛直方向）、軸回転速度

（２）潤滑油の温度

パラメータは、次の項目とする。

（１）ロータ系の重量

（２）軸受の種類

（３）軸受の負荷方向

#### (4) 不釣り合い量

パラメータは、ジャーナル部の軸直径の異なる軸を3種類、直径の異なるディスクを3種類、軸受長さの異なるスイングパッド軸受2種と1種類の真円軸受を用意した。またスイングパッド軸受は形状上負荷方向の違いにより、同じ物でも2種類のパラメータを取りうる。これらを交換し、繰り返し実験を行うものとする。

ディスクが変わると固有振動数が変わり、危険振動数も変わるため、それぞれ大きさ、重さの違うディスクを使用して振幅を測定することで、比較・検討を行うことができる。またスイングパッド軸受の長さも変わると、軸に作用するバネ合成も変化するので、振動や安定性も変わってくる。よって、スイングパッド軸受の長さを変えることで比較・検討することができる。スイングパッド軸受の負荷方向を変えることによりの LOP と LBP の違いにおいても比較検討を行うことができる。

真円軸受の実験は、同一条件での実際のスイングパッド軸受との比較を行うためのものである。

以下に実験手順を示す。

(1) 各装置の電源をいれる。次に、潤滑油用の給・排油ポンプを起動し、軸受ブロックに潤滑油が流れ込んでいることを確認する。駆動系のモータ固有振動回転数より低い回転数において徐々に上げながらしばらく廻す。潤滑油の温度が 30℃ 以上になってから測定を始める。

(2) 目的の回転速度になるよう軸の回転速度を合わせる。目的の回転速度で安定したと同時に、FFT アナライザを測定し、変位データをパーソナルコンピュータに取り込む。この時に、潤滑油の温度も記録しておく。

(3) 次の目的の回転速度に合わせ、(2)の動作を繰り返し測定を行う。

(4) 本実験においては、不釣り合い応答と安定限界速度を測定するため、軸の固有振動数より低い回転速度から軸の固有振動数の2倍以上の回転速度の範囲で実験を行う。具体的には、計測は 1500rpm ~

5500rpm まで行う。 本実験では、回転速度を100rpm ずつ増速し測定を行い、安定限界速度近辺においては 20rpm ずつ増速し測定を行う。引き続き次の実験を行う場合、潤滑油の温度が 35℃以下になってから行う。

(5) この方法により測定された結果を、パーソナルコンピュータにより FFT 処理を行い、考察する。

## 2-3 実験方法

### 2-3-1 非接触変位計の校正

(1) 正しい非接触変位計の校正を行う為には実際行う実験の装置を組み上げ、その装置を用いて校正を行わなければならない。今回の非接触変位計の校正は、実際に軸にディスクを取り付けた状態で行う。

次に、非接触変位計のプローブを非接触変位計の校正用のマグネットスタンドに取り付ける。プローブは軸方向と垂直に成っていることを確かめる。(プローブの先の面を軽くディスクに付けてみて隙間ができていないか見てみる。)

(2) 変換器、パーソナルコンピュータの電源をいれ、コンピュータを測定できるよう準備をする。準備ができれば、適当にプローブの高さと距離を決める。次に、適当に高さを動かしながら非接触変位計から出力された電圧値を読む。この時一番小さい電圧値になった時が最適な高さとなる。この位置はプローブが軸に水平になった位置である。位置が決まると、次はプローブをディスクに付かない程度まで近づける。

(3) 非接触変位計を等間隔づつはなす。その時の電圧値と距離を記録する。今回は  $50\mu\text{m}$  づつ 50 箇所を測定した。この時、注意すべきことは動かしすぎても戻さないようにする。

(4) 本実験では X-Y 方向に 2 カ所に非接触変位計を使用するため、(1)～(4)の手順をもう一方のチャンネルにおいても同様に行う。これらの結果を表2-6 (a),(b) と図2-11 (a),(b) に示す。この結果

より電圧と距離の関係を求め、その式を本実験のロータ変位の測定に使用した。

### 2-3-2 軸の固有振動数測定

測定したい軸の両端を自由端として固定する。 軸に圧電式加速度型振動ピックアップを軸に両面テープで取り付ける。 この振動ピックアップを振動計に接続し振動計の出力を FFT アナライザに取り付ける。 軸をハンマで軽く叩くことによりホワイトノイズを発生させる。 加振された軸は固有振動数で共振するため FFT アナライザの波形出力には固有振動数近辺で大きくなる。 この時のピークを示す周波数が軸の固有振動数である。 以上により求めた軸の固有振動数の結果を表2-7 に示す。

### 2-3-3 熱電対の温度校正

前項で書いたようにポータブルハイブリットレコーダは内部に校正回路が内蔵されているため、冷接点は必要としない。 しかしながら、正しい温度を得るために温度校正を行う。 その温度校正は、以下のように行う。 水槽にお湯を入れ、その中に潤滑油の入ったメスシリンダを入れる。 お湯と潤滑油の中にアルコール棒温度計を入れる。 お湯と潤滑油が目的の温度になり、かつそれぞれの温度差が小さくなった時に

ハイブリッドレコーダの値を読む。この動作を数回繰り返す。この時の熱電対と、それぞれの温度計の読みの値を表2-8に示す。

## 2-3-4 潤滑油の粘度と比重の測定

水槽にお湯を満たし、潤滑油を入れたメスシリンダをその中に入れる。

そのメスシリンダの中に、ウベローテ型粘度計、アルコール棒温度計、比重計を入れる。前項同様に水槽のお湯を目的の温度に近づけ、お湯と潤滑油の温度差が小さくなった時、その時の温度、比重、粘度を求める。比重は比重計の値を読み記録しておく。粘度の測定は以下のように行う。ウベローテ型粘度計は一定量の潤滑油を高い位置に引き上げ細い管を降りてくる時間経過を測定する。この値に定数を掛けると動粘度が求まる。今回用いたウベローテ型粘度計の定数は 0.00571 である。この求めた動粘度に比重を掛けることにより潤滑油の粘度を求めることができる。潤滑油の粘度の温度のデータを用い最小二乗法より、潤滑油の粘度と温度の関係式を求める。この式を用いて実験の潤滑油の粘度を求める。この時の、温度、比重、動粘度、粘度、最小二乗法による式の値を表2-9と図2-12に示す。

## 2-3-5 ロータ系の不つり合い除去

ロータ系はディスクと軸により構成されているが、ディスクの製作上、材質上の不つり合いが実験に大きな影響を及ぼすと思われる。

これを取り除くため、ディスクにあらかじめ設けたバランス取り用の穴にセットスクリュー（M4：各種の長さを用意）を取り付けることによって、危険速度通過時の振幅を抑えることにする。

以下にその手順を述べる。

（１）ディスクはセットスクリュー 8 本で軸に取り付けられているが、この 8 本のセットスクリューの締め付け具合を調節し、ディスク表面に指針を当てたダイヤルゲージを用いて静的変位を数十  $\mu\text{m}$  におさえる。

（２）図2-7に示すディスク側面のつりあいおもり穴のうち、90 度おきに任意の 4 箇所を選び、それぞれの位置についてつりあいおもりを取り付け、応答曲線を描かせ振幅の小さい 2 箇所を選定する。

（３）得られた 2 箇所の間の 2 つの穴についても、同様な作業を繰り返し、危険速度通過時の振幅が最小になる所を見いだす。

（４）1 箇所のバランスおもりで不足な場合は、その周囲の穴にもおもりを加え、最良の状態が見つかるまで、（２）と（３）の作業を繰り返す。

本来ならば、危険速度通過時の振幅をゼロにおさえるべきものなのであるが、本実験では上記の方法を用いて得られた最適の条件、すなわち、危険速度におけるロータの振幅が最小になる条件を採用する。



## 2-3-6 測定、観測・記録方法

本実験では、次の項目を測定する。

- (1) ロータの振幅（水平・鉛直方向）、軸回転速度
- (2) 潤滑油の温度

パラメータは、次の項目とする。

- (1) ロータ系の重量
- (2) 軸受の種類
- (3) 軸受の負荷方向
- (4) 不釣り合い量

パラメータは、ジャーナル部の軸直径の異なる軸を3種類、直径の異なるディスクを3種類、軸受長さの異なるスイングパッド軸受2種と1種類の真円軸受を用意した。またスイングパッド軸受は形状上負荷方向の違いにより、同じ物でも2種類のパラメータを取りうる。これらを交換し、繰り返し実験を行うものとする。

ディスクが変わると固有振動数が変わり、危険振動数も変わるため、それぞれ大きさ、重さの違うディスクを使用して振幅を測定することで、比較・検討を行うことができる。またスイングパッド軸受の長さも変わると、軸に作用するバネ合成も変化するので、振動や安定性も変わってくる。よって、スイングパッド軸受の長さを変えることで比較・検討することができる。スイングパッド軸受の負荷方向を変えることによりのLOPとLBPの違いにおいても比較検討を行うことができる。

真円軸受の実験は、同一条件での実際のスイングパッド軸受との比較を行うためのものである。

以下に実験手順を示す。

(1) 各装置の電源をいれる。次に、潤滑油用の給・排油ポンプを起動し、軸受ブロックに潤滑油が流れ込んでいることを確認する。駆動系のモータ固有振動回転数より低い回転数において徐々に上げながらしばらく廻す。潤滑油の温度が 30℃以上になってから測定を始める。

(2) 目的の回転速度になるよう軸の回転速度を合わせる。目的の回転速度で安定したと同時に、FFT アナライザを測定し、変位データをパーソナルコンピュータに取り込む。この時に、潤滑油の温度も記録しておく。

(3) 次の目的の回転速度に合わせ、(2)の動作を繰り返し測定を行う。

(4) 本実験においては、不釣り合い応答と安定限界速度を測定するため、軸の固有振動数より低い回転速度から軸の固有振動数の 2 倍以上の回転速度の範囲で実験を行う。具体的には、計測は 1500rpm ~ 5500rpm まで行う。本実験では、回転速度を100rpm ずつ増速し測定を行い、安定限界速度近辺においては 20rpm ずつ増速し測定を行う。引き続き次の実験を行う場合、潤滑油の温度が 35℃以下になってから行う。

(5) この方法により測定された結果を、パーソナルコンピュータにより FFT 処理を行い、考察する。

実験に取ったパラメータを以下に示す。

1) 真円軸受、 $L/D=1 \cdot LOP$ 、 $LBP$ の比較

data 1 : 軸1 Disk 1  
 data 2 : 軸2 Disk 1  
 data 3 : 軸3 Disk 1  
 data 4 : 軸2 Disk 1  
 data 5 : 軸2 Disk 2  
 data 6 : 軸2 Disk 2  
 data 7 : 軸2 Disk 3  
 data 8 : 軸3 Disk 2  
 data 9 : 軸3 Disk 3

2) 軸1、2の比較

data 10 :  $L/D=1$  LOP Disk 1  
 data 11 :  $L/D=1$  LBP Disk 3  
 data 12 :  $L/D=1$  LBP Disk 1  
 data 13 :  $L/D=1$  LBP Disk 2

3)  $LOP$ 、 $LBP$ の比較

data 14 :  $L/D=1$  軸1 Disk 1  
 data 15 :  $L/D=0.75$  軸1 Disk 1  
 data 16 :  $L/D=1$  軸1 Disk 1  
 data 17 :  $L/D=1$  軸1 Disk 2  
 data 18 :  $L/D=0.75$  軸1 Disk 2  
 data 19 :  $L/D=0.75$  軸2 Disk 1  
 data 20 :  $L/D=1$  軸1 Disk 3  
 data 21 :  $L/D=0.75$  軸1 Disk 3  
 data 22 :  $L/D=1$  軸2 Disk 1  
 data 23 :  $L/D=1$  軸3 Disk 1

#### 4) L/Dの比較

data 24 : LOP 軸1 Disk 1  
 data 25 : LBP 軸1 Disk 2  
 data 26 : LBP 軸2 Disk 1  
 data 27 : LBP 軸2 Disk 2  
 data 28 : LBP 軸1 Disk 1  
 data 29 : LOP 軸1 Disk 2  
 data 30 : LBP 軸2 Disk 1  
 data 31 : LBP 軸2 Disk 3  
 data 32 : LBP 軸1 Disk 3

#### 5) 負荷による比較

data 33 : L/D=1 LOP 軸1  
 data 34 : L/D=1 LBP 軸3  
 data 35 : L/D=1 LBP 軸2  
 data 36 : L/D=1 LBP 軸1  
 data 37 : L/D=0.75 LBP 軸2  
 data 38 : L/D=0.75 LBP 軸1  
 data 39 : L/D=0.75 LOP 軸1

#### 6) 不釣り合い応答 座金 0, 2, 4

data 40 : L/D=0.75 LBP 軸2 Disk 2  
 data 41 : L/D=0.75 LBP 軸1 Disk 1  
 data 42 : L/D=1 LBP 軸3 Disk 1  
 data 43 : L/D=1 LOP 軸1 Disk 1  
 data 44 : L/D=1 LBP 軸1 Disk 1  
 data 45 : L/D=1 LBP 軸1 Disk 1

## 第3章 結果と考察

本章では実験によって計測された、固有振動数近辺の回転速度における不釣り合い応答振幅（mm）と安定限界速度（rpm）を結果として図に示し、スイングパッド軸受の性能を考察する。また、真円軸受との性能も比較していく。考察を行うパラメータを以下に示す。

- 1) 不釣り合い量
- 2) スイングパッド軸受の負荷方向（LOP,LBP）
- 3) 軸受幅（L/D）
- 4) 軸重量
- 5) 軸受半径隙間（C/D）

本研究では軸受を3種類（内1種類は真円軸受）、軸を3本、円板を3種類用意した。これによってとれる実験パラメータは、軸受幅に2種類、軸受半径隙間に2種類、軸重量に3種類、偏重心に3種類である。

その中より必要と思われる組合せを選び出し、軸の回転速度を1500rpm-5500rpmまで回し、上で述べた不釣り合い応答振幅と危険回転速度を求めた。

### 3-1 不釣り合い応答

図3-1 から図3-3 までにスイングパッド軸受の不釣り合い応答の結果を示す。また、図3-4 に軸に不釣り合いを与えたときの安定限界速度の結果を示す。これらの図より、軸に加える不釣り合いの量が増加しても安定限界は変わらず、固有振動数近辺の回転速度における振動振幅は大きくなることがわかる。

つぎに、図3-1 から図3-2 までにおいて、 $L/D=0.75$  と  $L/D=1$  の不釣り合い振動振幅についてみると、前者の方が、明らかに振動振幅は大きい。

また、 $L/D=1$  の時のこの比較は、図3-2 と図3-3 より LOP の方が明らかに振動振幅が大きいことがわかる。これらより、 $L/D=1$  より  $L/D=$

0.75の方が、LBPよりLOPの方が、それぞれ軸受の剛性が高いことが言える。

次に、図3-2から図3-3においては、振動振幅の最大値を持つ回転速度は、LOPの方が回転速度が高くなることがわかる。これは、前章で、計測した、軸本来の固有振動数近くに最大振動振幅を持つ回転速度が移行したためであると考えられる。このことは、ティルティングパッド軸受においてみられる傾向と同じである。

また、本研究では真円軸受における、不釣り合い応答についてのデータは取ってはいないが、他の研究における真円軸受についてのデータによると真円軸受の方が軸の固有振動数近辺での振動振幅がスイングパッド軸受におけるものよりもなだらかになっている。このことは、真円軸受の方が、スイングパッド軸受より軸の固有振動数近辺の回転速度において安定した性能を得られることができると考えられる。

### 3-2 負荷方向の比較

図3-5に、LBPとLOPとで軸を支持した場合の軸の固有振動数近辺の回転速度と振動振幅との関係を比較したものを示し、図3-6に、LBPとLOPの安定限界速度の比較したものを示す。図3-5よりわかるように、軸の固有振動数近辺での軸の振動振幅は、LBPよりLOPの方が振動振幅が大きいことがわかる。この事より、LOPの方がLBPに比べ軸受の剛性は大きいと考えられる。また、図3-6よりLOPの方が、LBPに比べ安定限界が高い。この事より、高回転域を運転する場合においては、LBPより、LOPの方が軸受としては安定した性能を得られ、また軸の固有振動数近辺の軸速度においては、LOPよりLBPの方がより安定した性能が得られると考えられる。

### 3-3 軸受幅による比較

図3-7 までは軸受幅の異なるスイングパッド軸受における、固有振動数近辺の振動振幅を比較した結果を示す。また図3-8 までは、軸受幅の異なる軸受における安定限界速度の比較を示している。

図3-7 までの図を見ると、明らかに、 $L/D=1$  より  $L/D=0.75$  の方が、固有振動数近辺の最大振動振幅が大きいことがわかる。また、 $L/D=1$  の場合、安定した振幅値を示し、 $L/D=0.75$  場合は振幅値は、双峰性を示し、あまり安定した動作は見られない。

次に、図3-8 までの結果から、 $L/D=1$  より  $L/D=0.75$  の方が、安定限界が高いと考えられる。

これらの事より、高回転域を使用する場合に於いては  $L/D=0.75$  の方がより安定した性能が得られる。

### 3-4 軸重量の違いによる比較

図3-9 まではスイングパッド軸受で異なる軸重量の軸を支持した場合の固有振動数近辺の振動振幅の結果を示す。また図3-10 は、軸重量の異なる場合の軸系の安定限界回転速度の結果を示す。軸重量はディスクの重さで調整され、Disk1、Disk2、Disk3 の順で重くなっていく。

図3-9 の結果より、明らかに軸重量の増加とともに固有振動数が下がっていることがわかる。振幅振動が変化するのは、ディスクを取り付けたときに、偏重心の位置及び大きさが変わってしまったからと考えられるので、振動振幅の大きさはここでは考察を行はない。

つぎに、図3-10 の安定限界速度の結果において、ディスクの種類が少ないため、今回の実験では、はっきりとした結果を得ることは出来なかったが、明らかに、軸重量の増加に伴い安定限界速度が低下していることがわかる。

### 3-5 軸受半径隙間の違いによる比較

図3-11 までにスイングパッド軸受の軸受半径隙間の異なる軸受で軸を支持した場合の軸系の固有振動数近辺の振動振幅の結果を示す。また図3-12 は、軸受半径隙間の異なる場合の安定限界速度の結果を示す。また、第2章で述べたように、軸2より、軸3の方が軸半径が小さくなっているため、よって、軸3の方が軸受半径隙間が大きいことになる。図3-11 までを見ていくと、軸2より軸3の方が最大振動振幅が小さいことがわかる。これは軸受半径隙間が小さい場合の方が軸を剛に支持していることを意味する。

次に、図3-12 の軸に対する、安定限界速度について考察する。図3-12 の結果において、軸1は軸自体の半径が $\phi 10$ であり、他の軸2、3の $\phi 12$ とは異なるため比較することは出来ないが、参考のため記入した。

これらの図から、軸2より、軸3の方が、安定限界速度が低いことがわかる。

このことより、軸受半径隙間がより小さい方が高回転域を利用する場合には、より安定した性能を得ることができる。

### 3-6 スイングパッド軸受と真円軸受の比較

図3-13 までにスイングパッド軸受と真円軸受とで軸系を支持した場合の固有振動数近辺の振動振幅を比較した結果である。また図3-14 は、スイングパッド軸受と真円軸受における安定限界速度を比較した結果である。

図3-13 において固有振動数近辺の回転速度では、スイングパッド軸受と真円軸受の軸受の振動振幅の形は似ているが、真円軸受の方が、振動振幅は大きいことが読み取れる。このことから、スイングパッド軸受の方が真円軸受より剛な支持であることがわかる。

また、図3-14 の結果から、軸受半径隙間における比較では、スイング



パッド軸受で軸系を支持した場合の方が、安定限界が高いことがわかる。

また、図3-14の結果から、軸重量の違いでの比較においても、スイングパッド軸受で軸系を支持した場合の方が安定限界が高いことがわかる。以上より、振動振幅はスイングパッド軸受の方が小さく、軸受剛性は真円軸受の方が大きくなるが、安定限界速度はスイングパッド軸受の方が大きくなっている。これは一見、矛盾した挙動に見えるが、スイングパッド軸受のパッド部の支持ゴムの振動減衰効果により安定限界速度が真円軸受における安定限界速度より大きくなっていると考えられる。これらの事より、高回転域に於いては、スイングパッド軸受の方が真円軸受より安定した性能が得られると考えられる。

表 3 - 1 各パラメータでの安定限界速度

L/D	状態	軸	Disk	座金	出現	消滅	FFT (rpm)
0.75	LBP	1	1	0	-	-	-
0.75	LOP	1	1	0	-	-	5000
0.75	LBP	1	1	2	-	-	3700
0.75	LBP	1	1	4	-	-	3800
0.75	LBP	1	2	0	-	-	3100
0.75	LOP	1	2	0	-	-	2600
0.75	LBP	1	3	0	-	-	4600
0.75	LOP	1	3	0	-	-	-
0.75	LBP	2	1	0	-	-	-
0.75	LOP	2	1	0	-	-	4200
0.75	LBP	2	2	0	5500	5450	5520
0.75	LBP	2	3	0	4820	4680	4820
1	LBP	1	1	0	-	-	2200
1	LBP	1	1	2	-	-	2200
1	LBP	1	1	4	-	-	2200
1	LBP	1	1	0	-	-	3200
1	LBP	1	1	2	-	-	3100
1	LBP	1	1	4	-	-	5000
1	LOP	1	1	0	-	-	4600
1	LOP	1	1	2	-	-	-
1	LOP	1	1	4	-	-	4200
1	LBP	1	1	0	-	-	4300
1	LBP	1	2	0	-	-	4000
1	LOP	1	2	0	-	-	4000
1	LBP	1	3	0	-	-	3900
1	LOP	1	3	0	-	-	5000
1	LBP	2	1	0	-	-	-

L/D	状態	軸	Disk	座金	出現	消滅	FFT (rpm)
1	LOP	2	1	0	-	-	4500
1	LOP	2	1	0	-	-	5700
1	LBP	2	2	0	5360	5320	-
1	LBP	2	3	0	4700	4720	4760
1	LBP	2	3	0	2280	-	5200
1	LBP	3	1	0	-	-	-
1	LBP	3	1	2	5260	5220	-
1	LBP	3	1	4	5600	5580	-
1	LOP	3	1	0	3160	-	-
1	LBP	3	2	0	5280	5220	5180
1	LBP	3	3	0	4580	4560	4580
1	真円軸受	1	1	0	-	-	4900
1	真円軸受	2	1	0	-	-	5500
1	真円軸受	2	2	0	5320	-	4220
1	真円軸受	2	1	0	-	-	4600
1	真円軸受	2	2	0	-	-	5300
1	真円軸受	2	1	0	-	-	4200
1	真円軸受	2	2	0	-	-	4100
1	真円軸受	2	3	0	-	-	5400
1	真円軸受	3	1	0	3220	3180	3200
1	真円軸受	3	2	0	3840	3820	5400

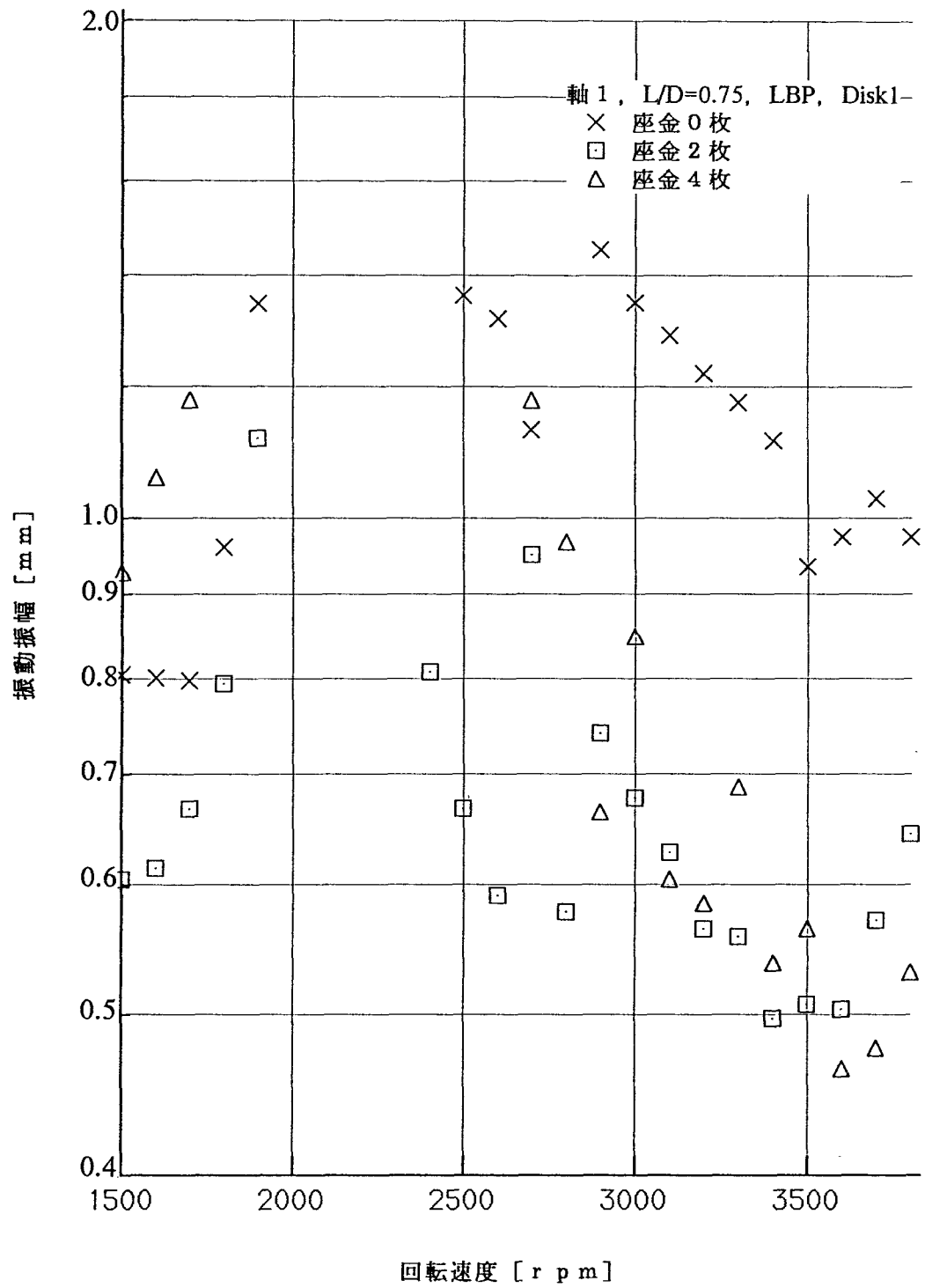


図 3 - 1 軸系の不釣り合い応答に対する不釣り合い量の影響

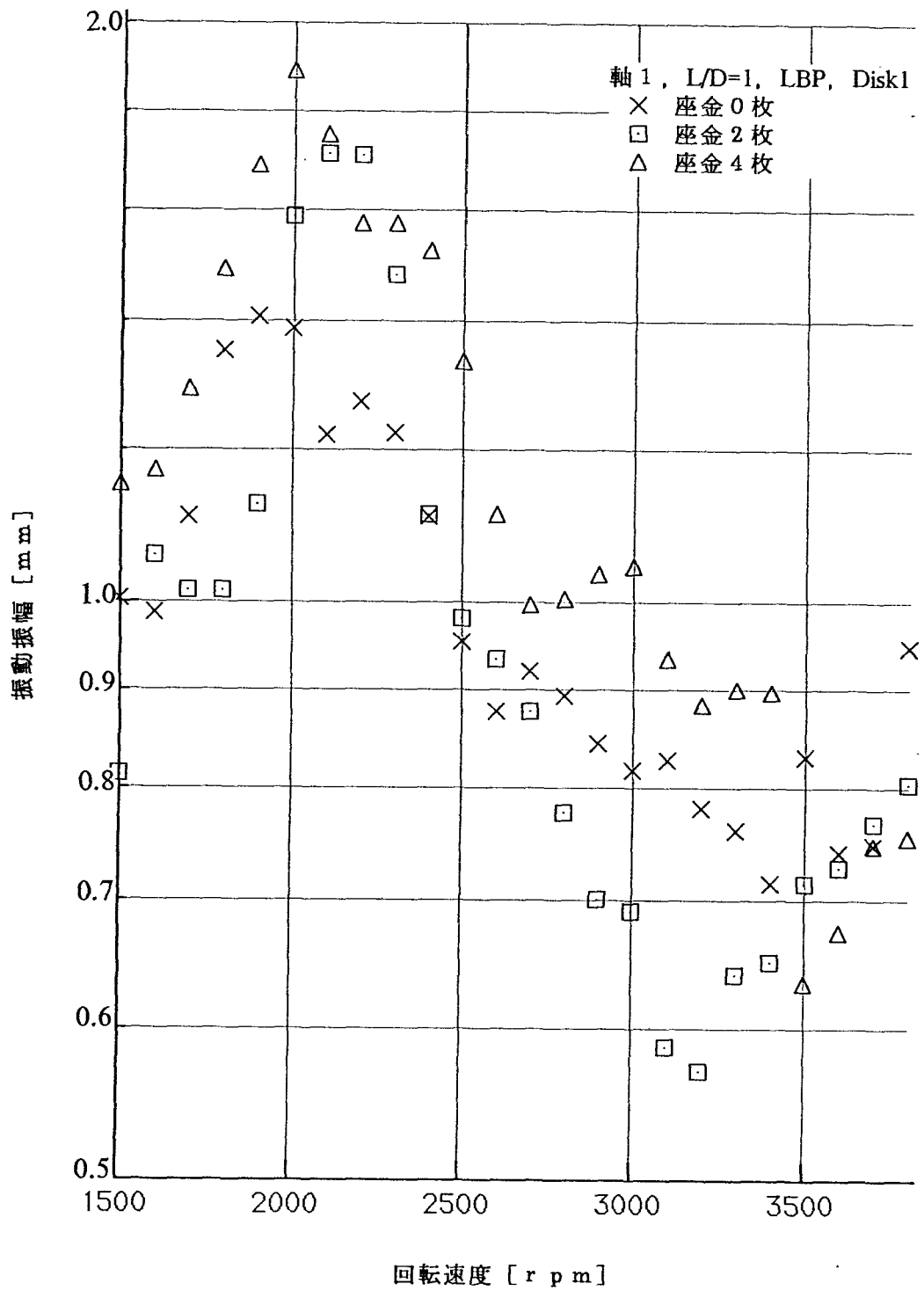


図 3 - 2 軸系の不釣り合い応答に対する不釣り合い量の影響

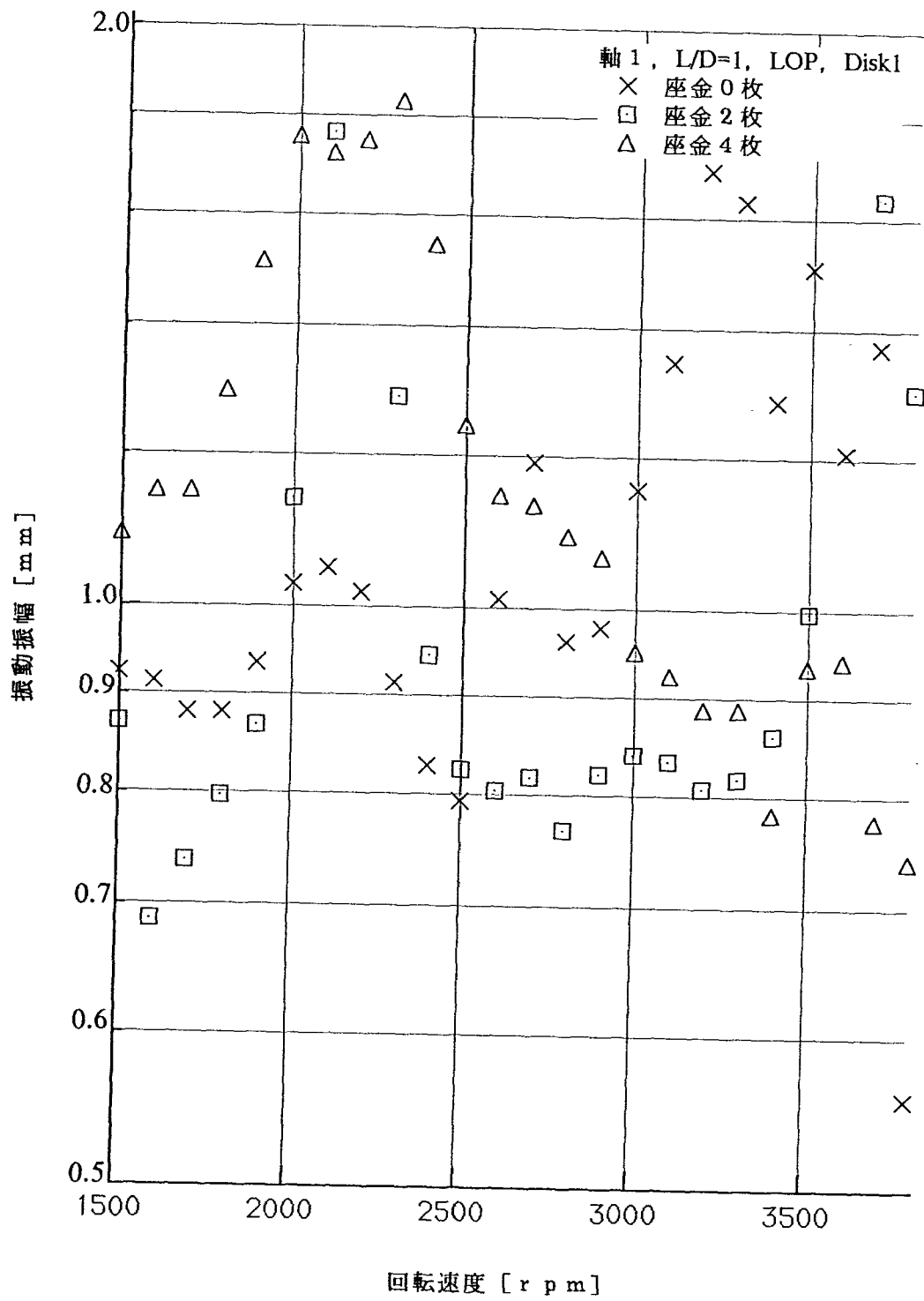


図 3 - 3 軸系の不釣り合い応答に対する不釣り合い量の影響

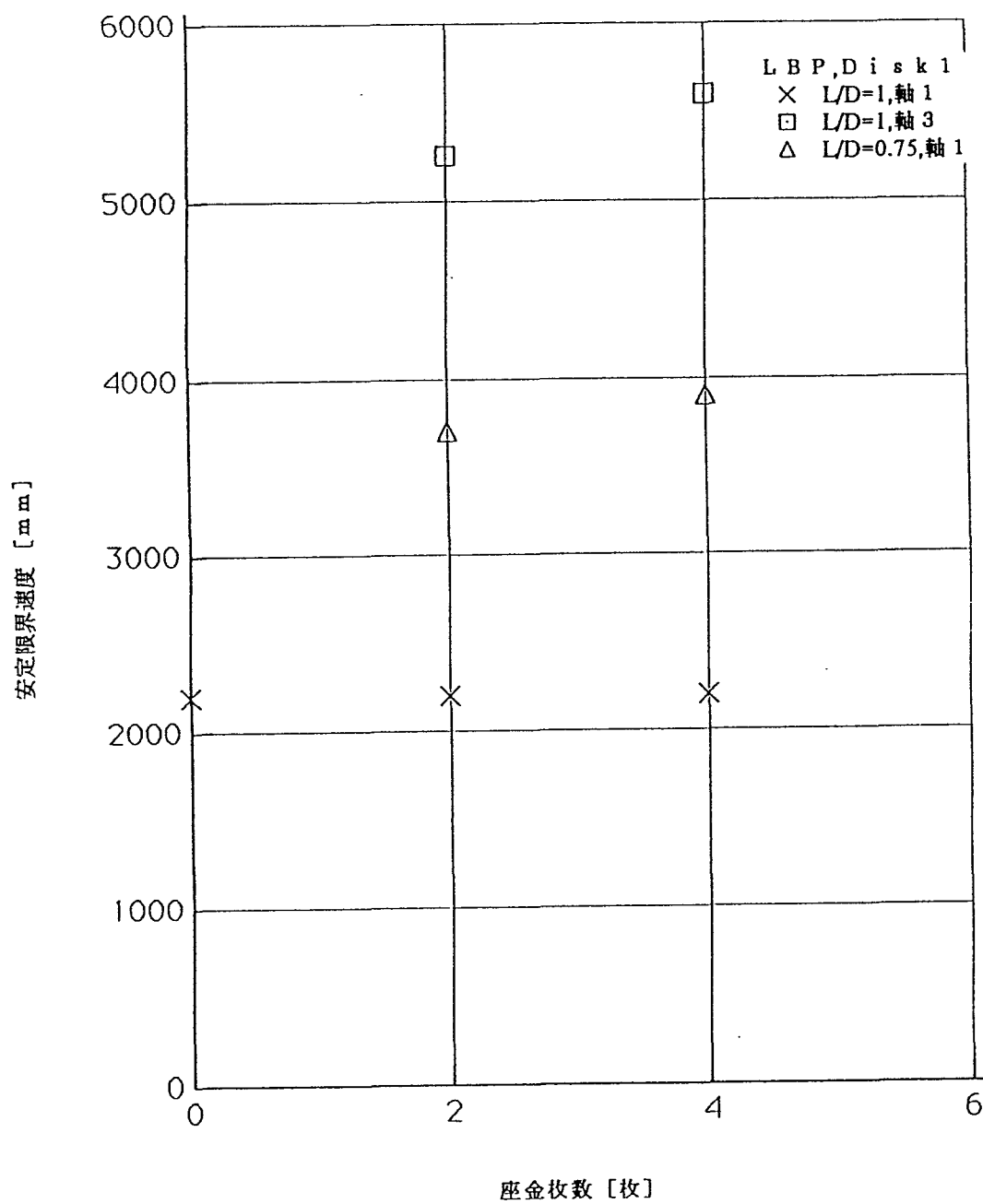


図 3 - 4 安定限界速度に対するの不釣り合い量の影響

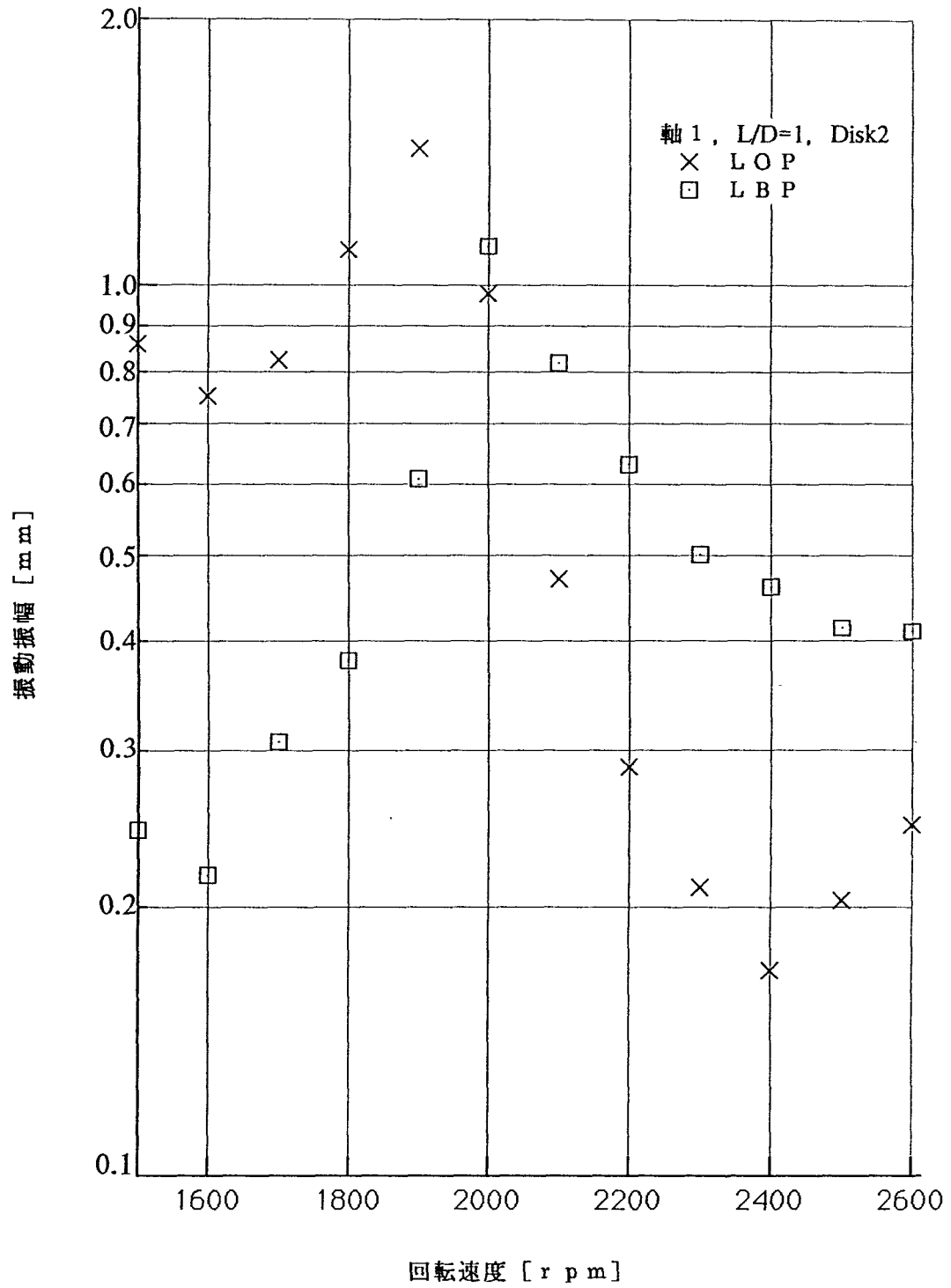


図 3 - 5 軸系の不釣り合い応答に対する軸受の負荷方向の影響



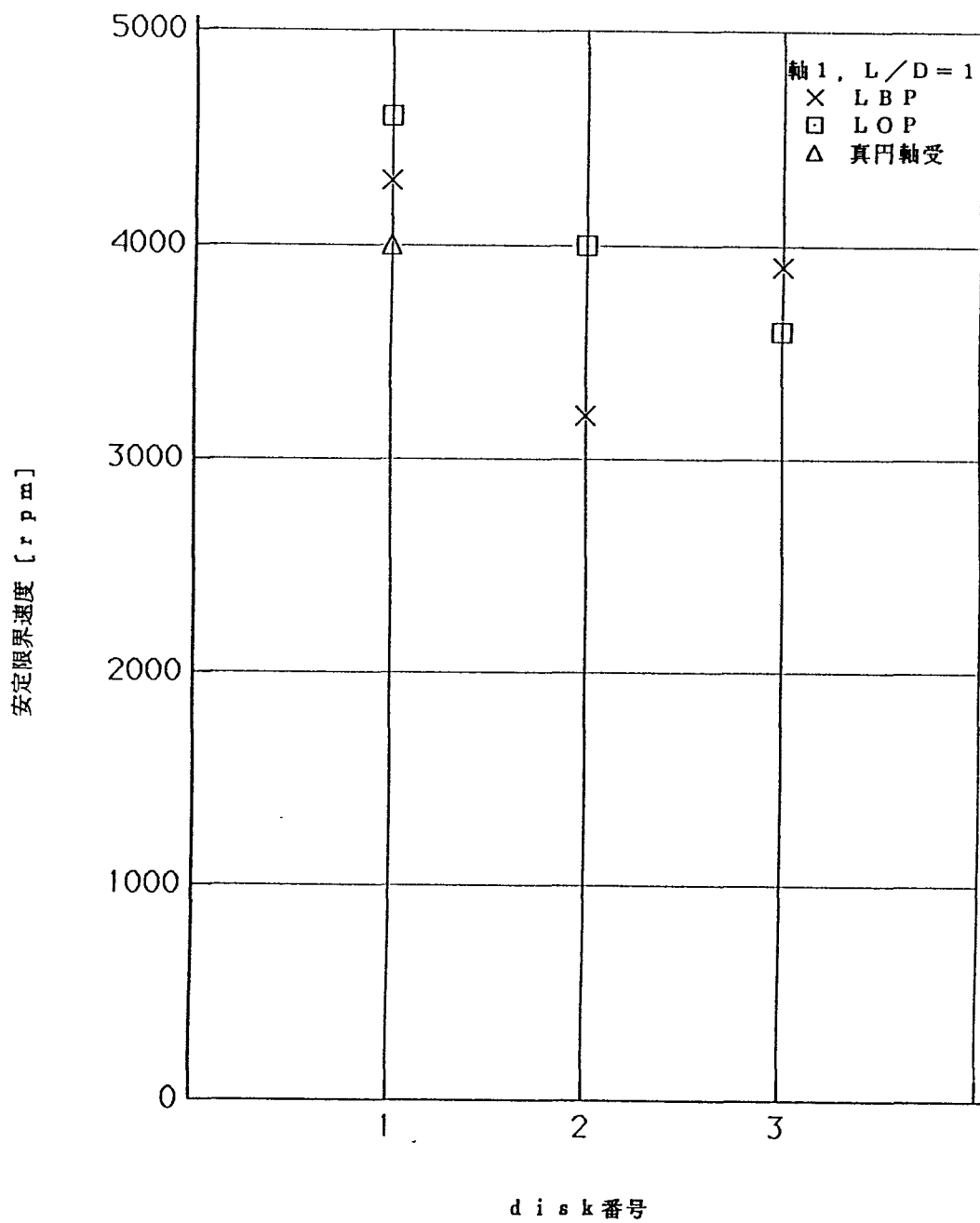


図 3 - 6 安定限界速度に対する軸受の負荷方向の影響

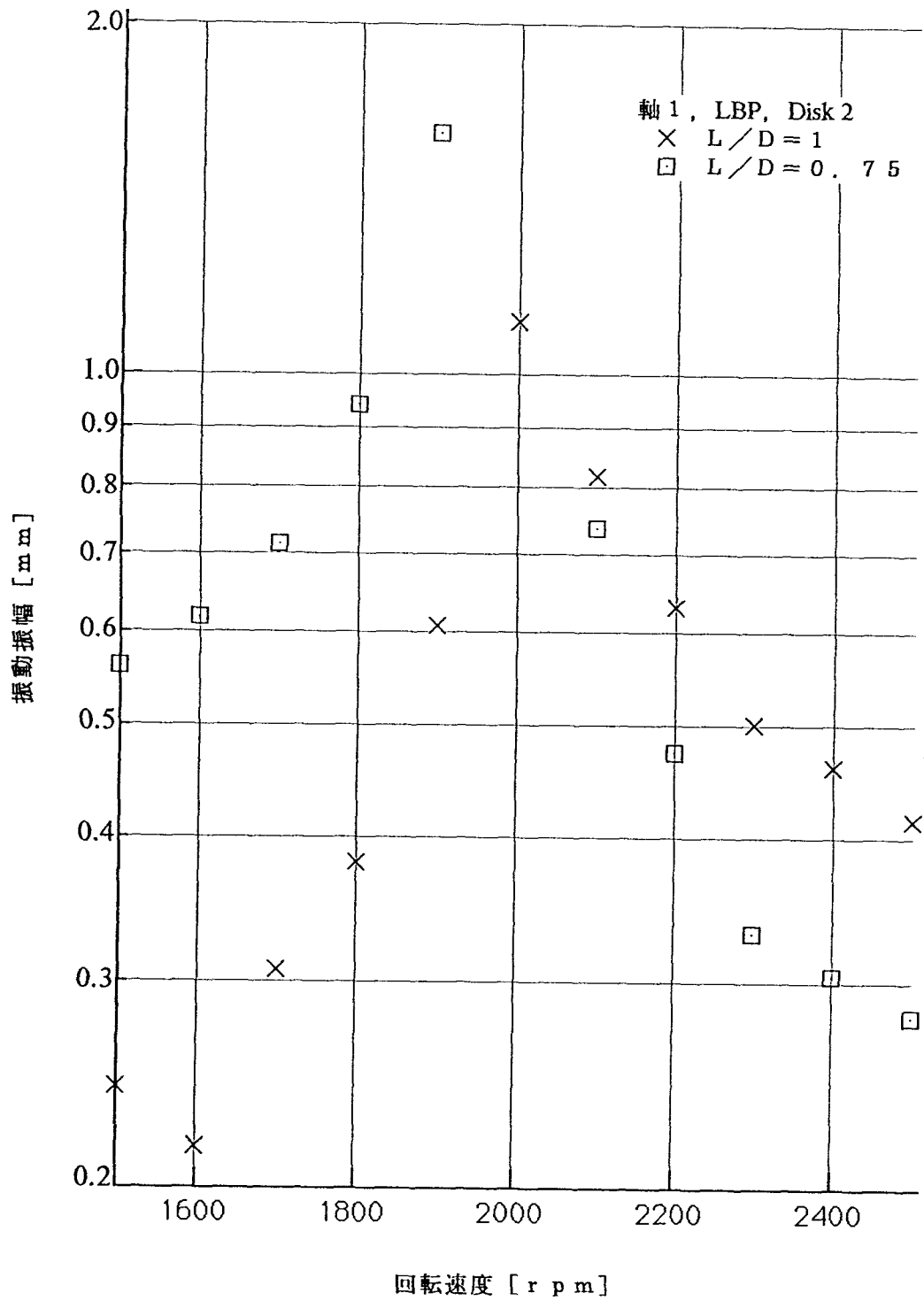


図 3 - 7 軸系の不釣り合い応答に対する軸受幅の影響

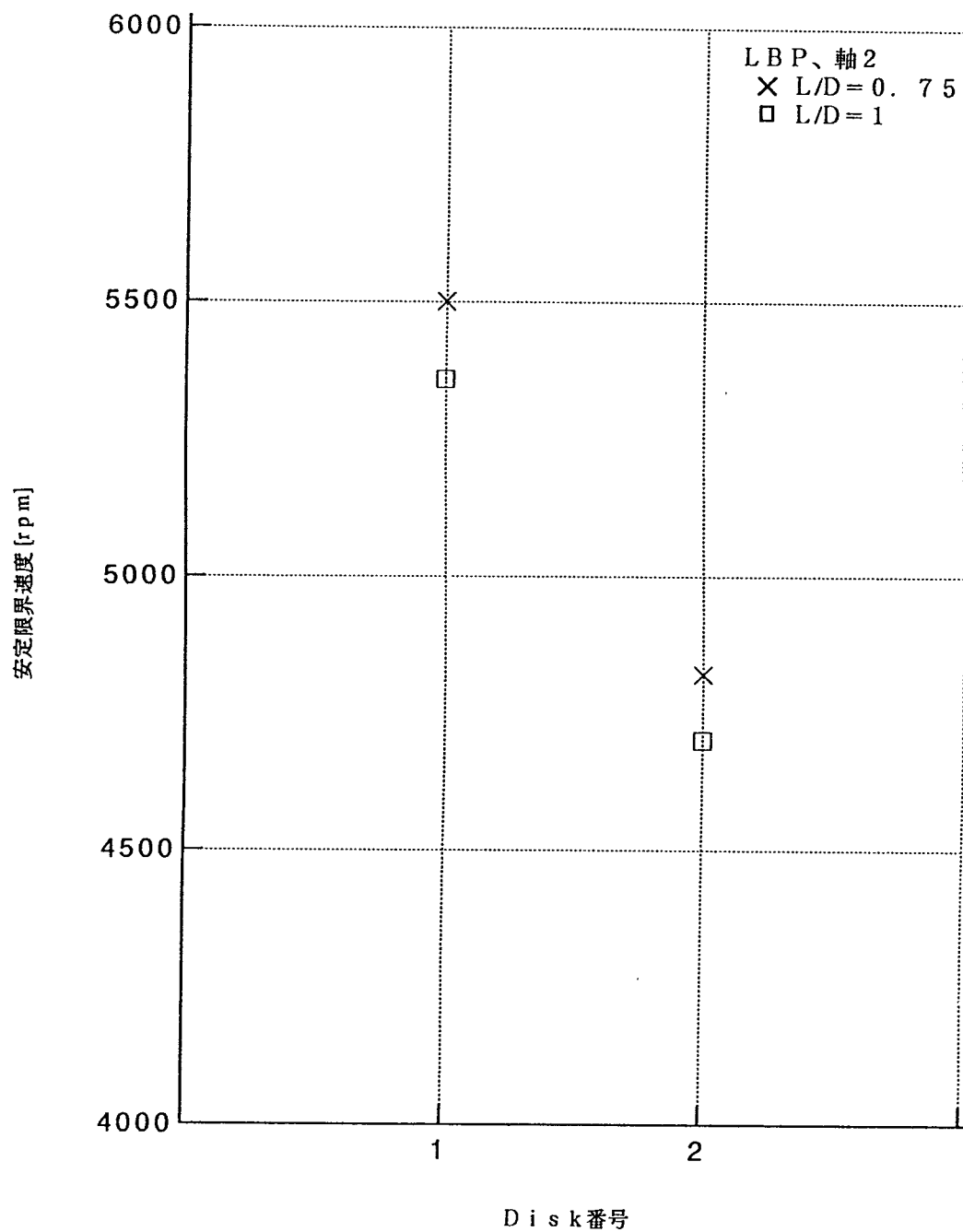


図 3 - 8 安定限界速度に対する軸受幅の影響

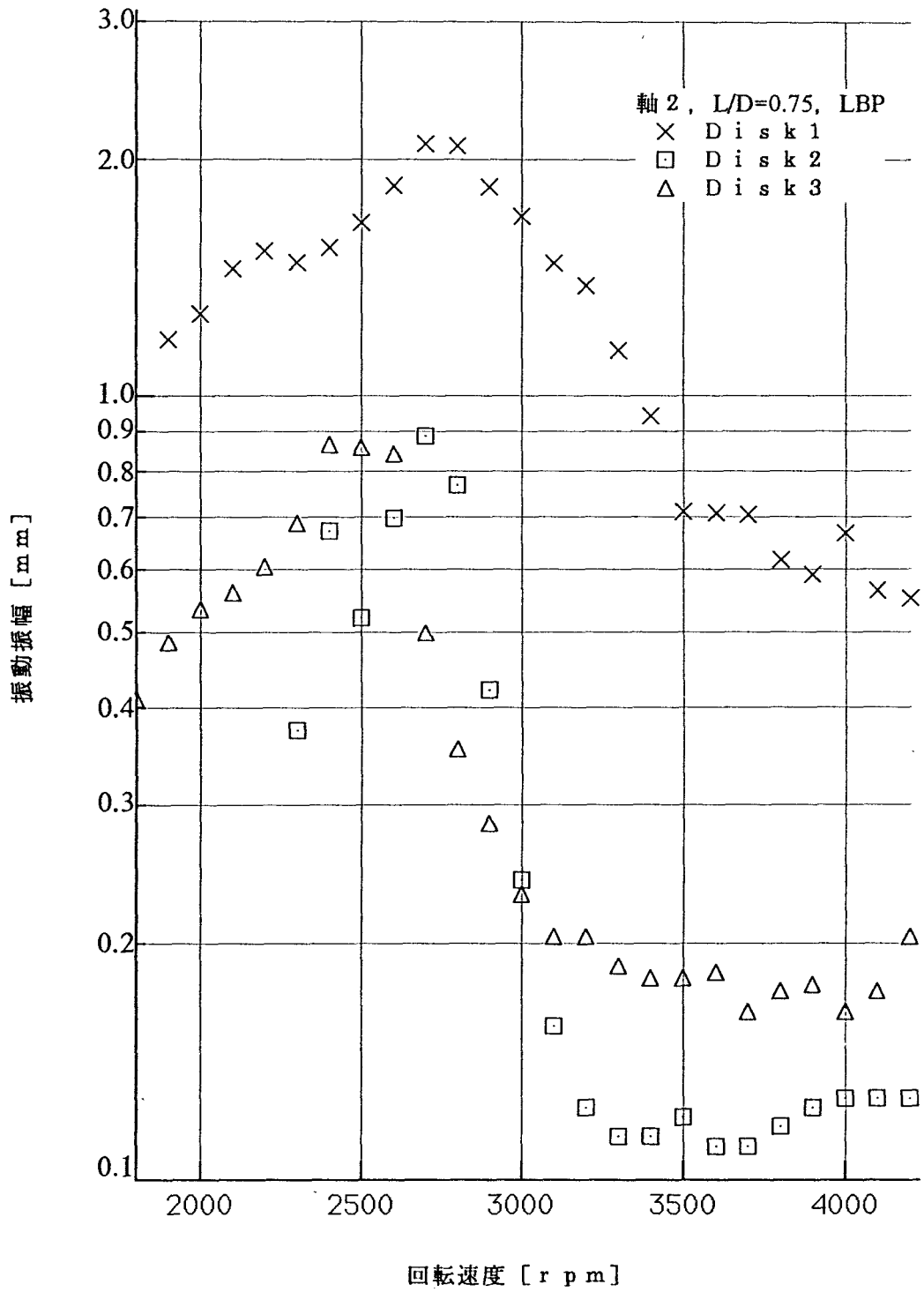


図 3 - 9 軸系の不釣り合い応答に対する軸重量の影響

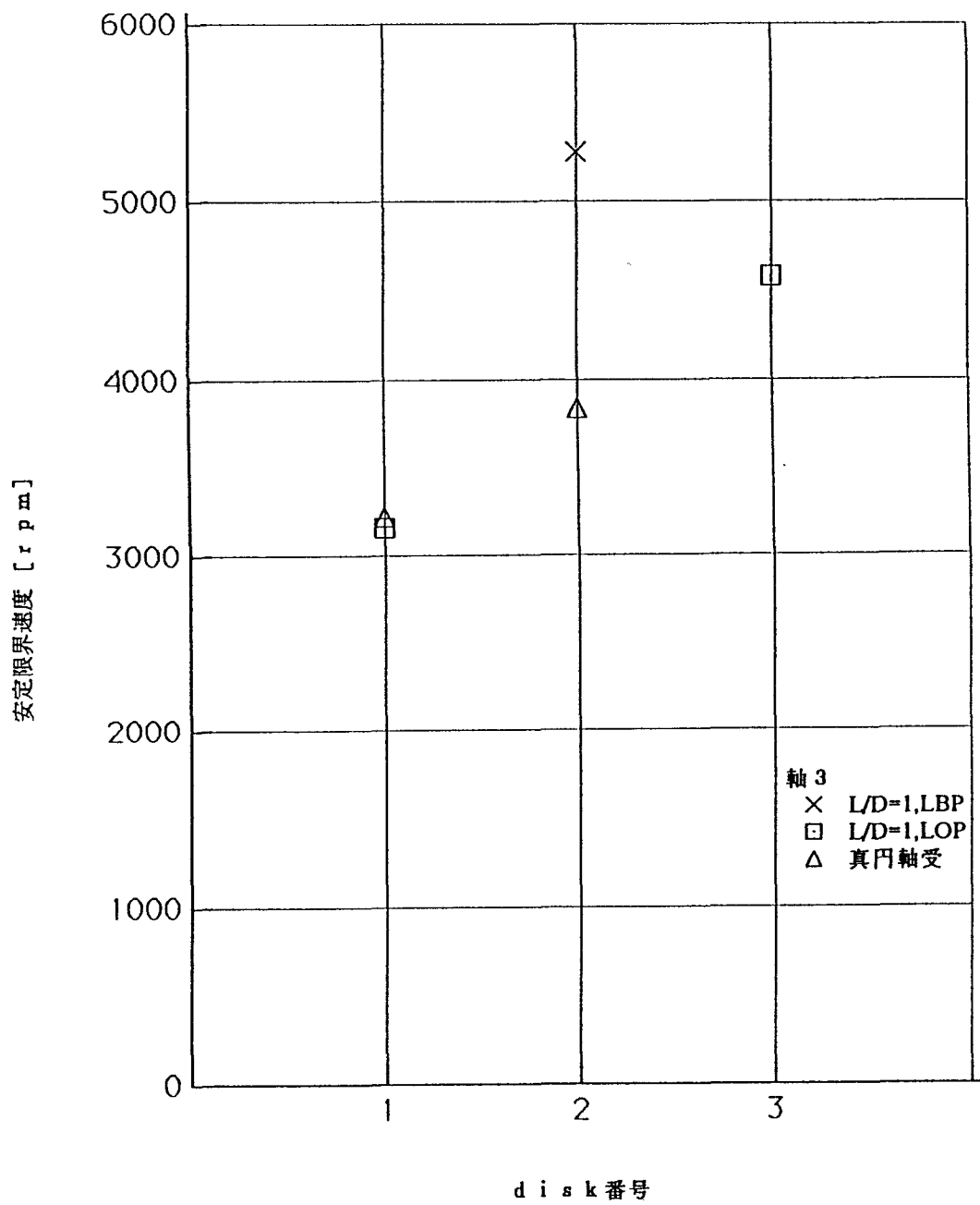


図 3 - 1 0 安定限界速度に対する軸重量の影響

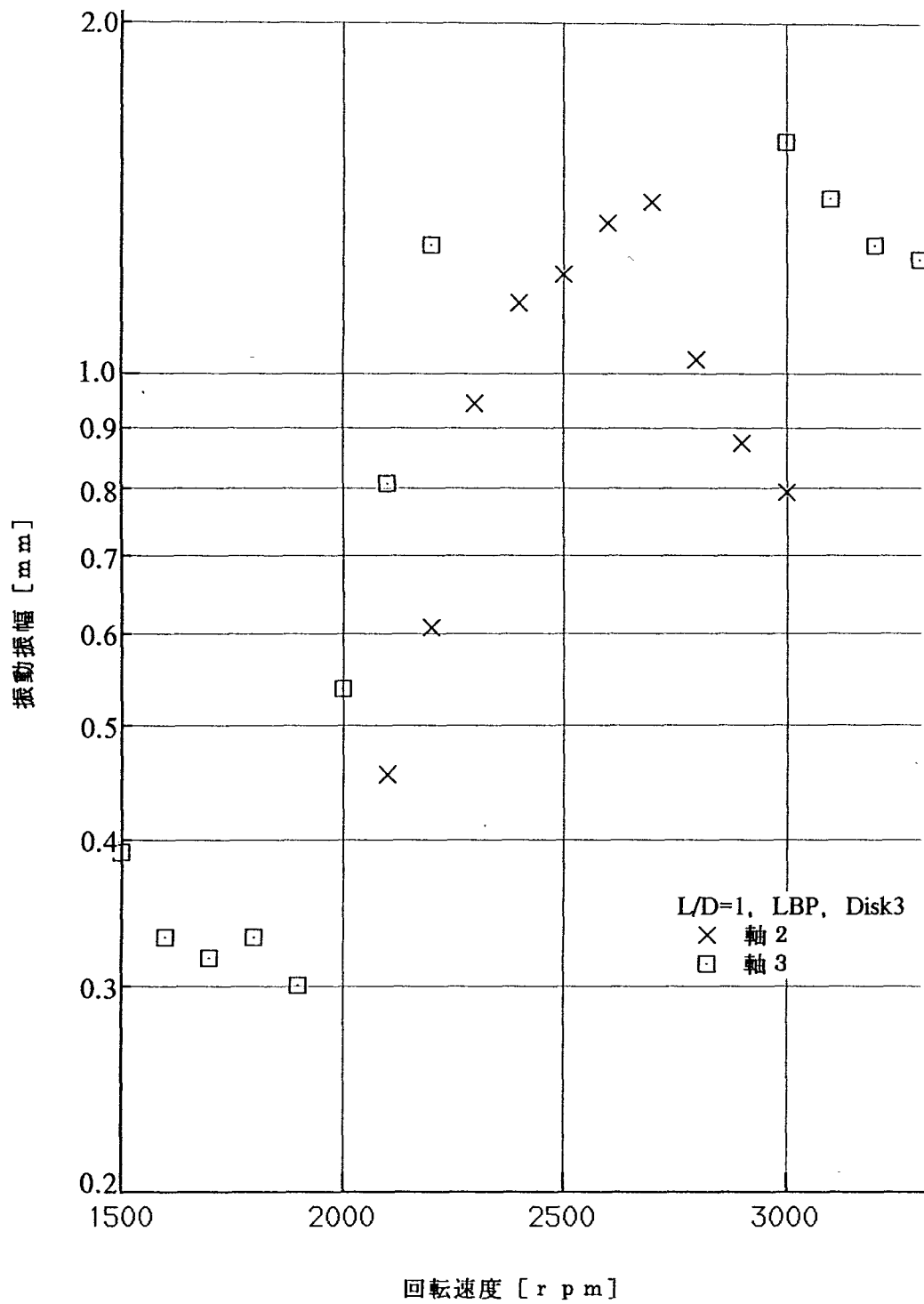


図 3 - 1 1 軸系の不釣り合い応答に対する軸受半径隙間の影響

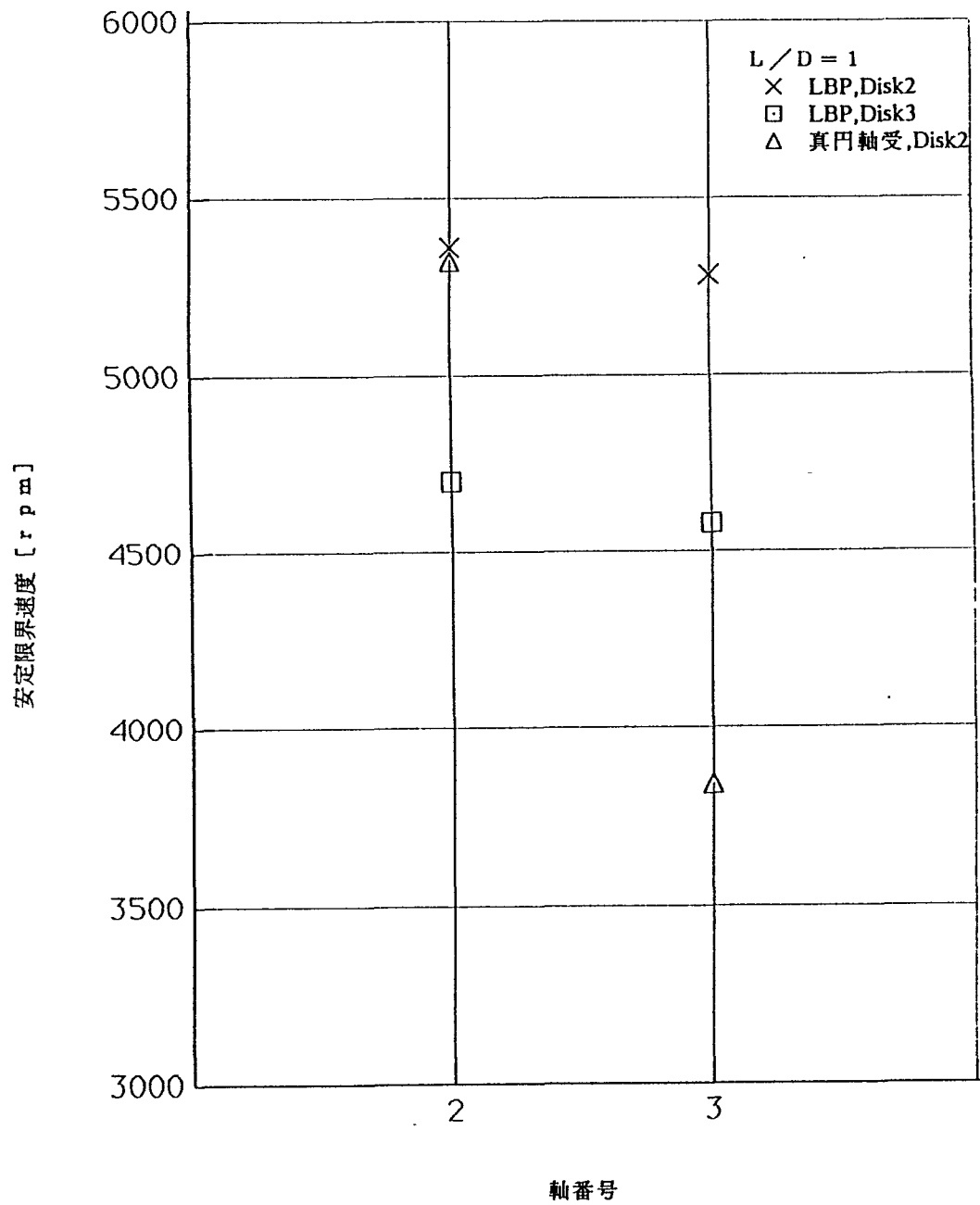


図 3 - 1 2 安定限界速度に対する軸半径隙間の影響

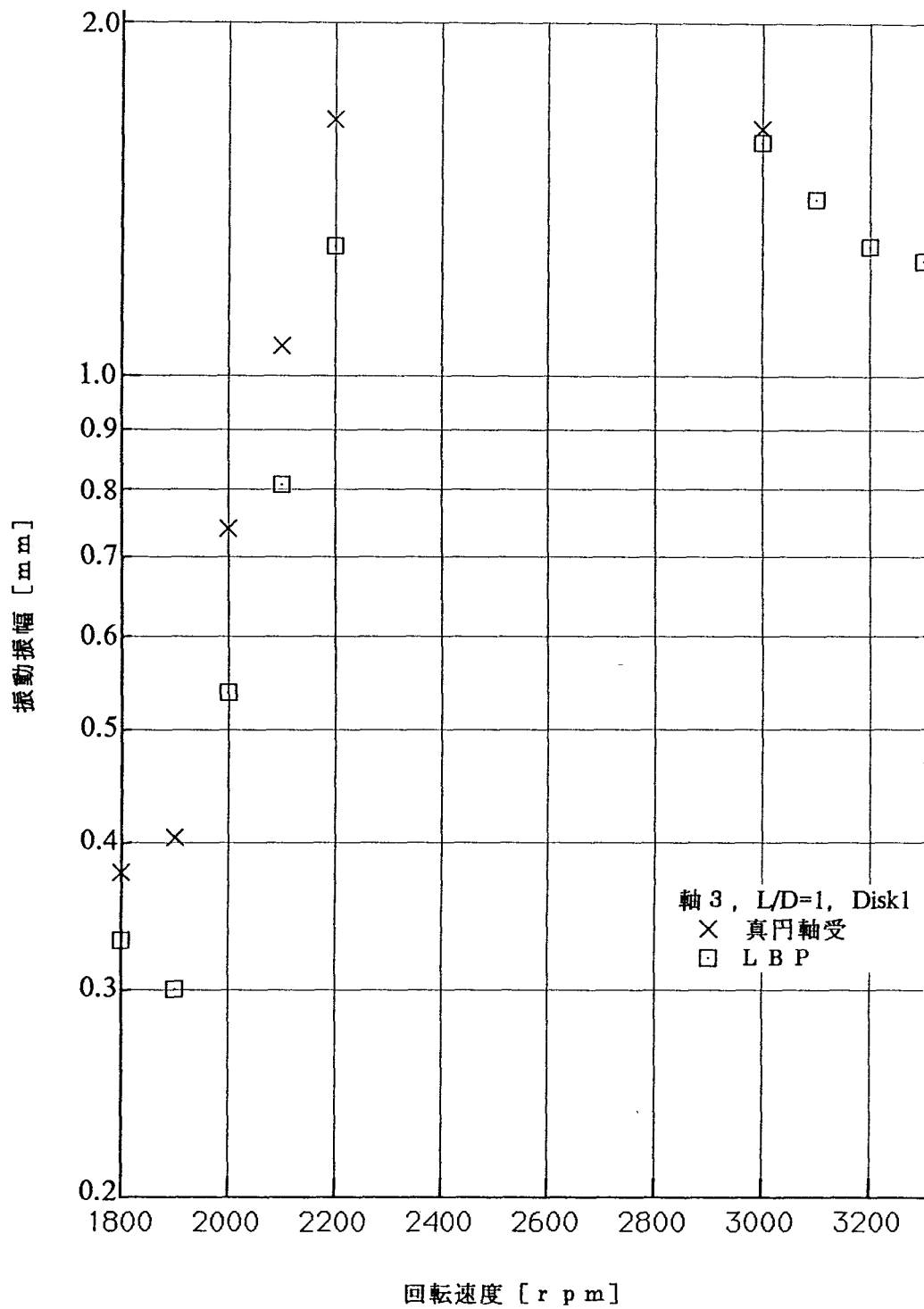


図 3 - 1 3 スイングパッド軸受と真円軸受における  
不釣り合い応答の比較



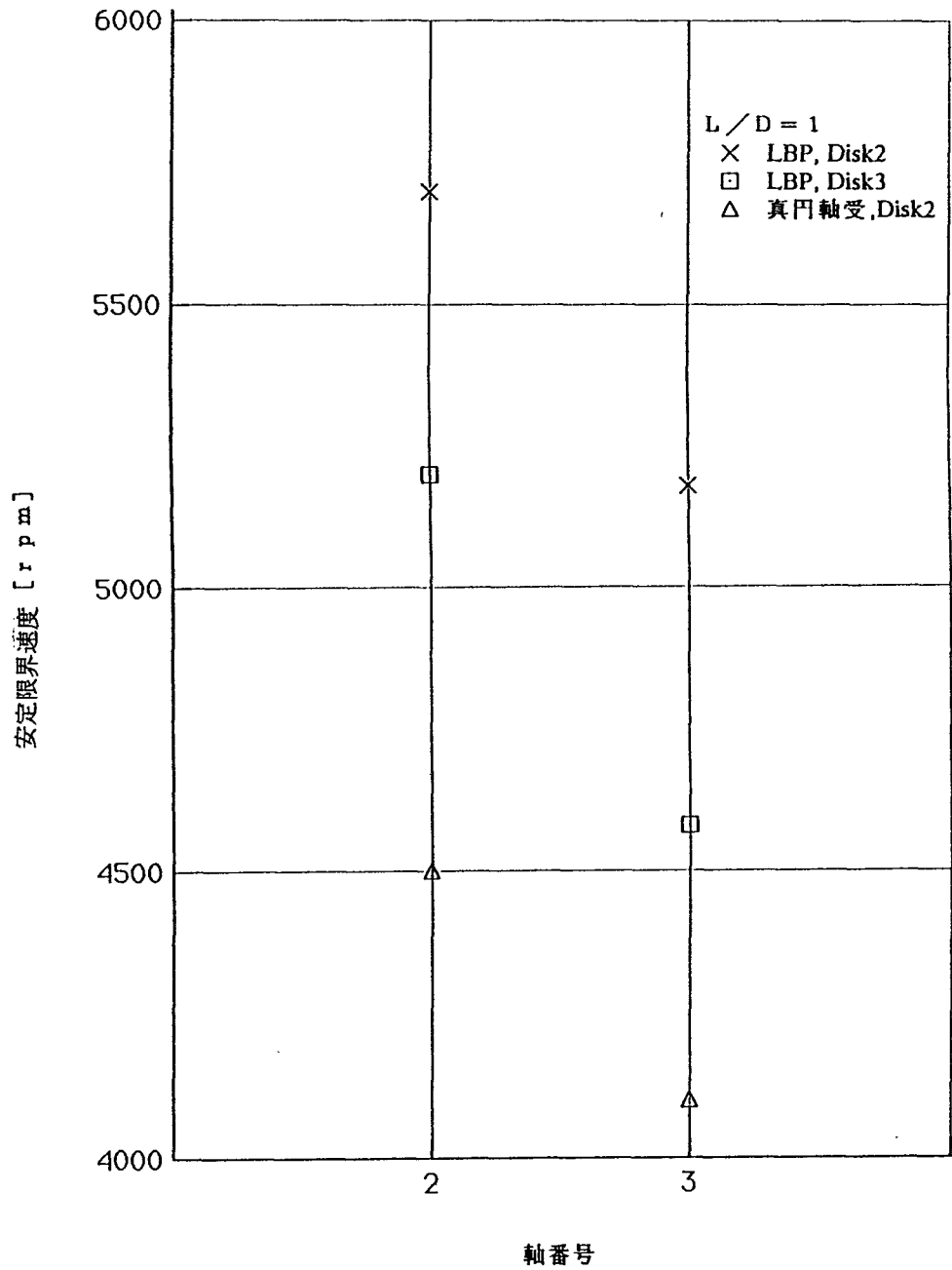


図 3 - 1 4 スイングパッド軸受と真円軸受における  
安定限界速度の比較

## 第4章 結 論

結果と考察をまとめると以下のようなこのがわかる。

- 1) LBP より LOP の方が、安定限界速度は高くなる。
- 2)  $L/D=1$  より  $L/D=0.75$  の方が安定限界速度は高くなる。
- 3) 軸重量の増加にともない安定限界速度は低下する。
- 4) 軸受半径隙間が小さい方が軸受剛性が高く、安定限界速度は高い。
- 5) スイングパッド軸受は真円軸受より安定限界は高く、また軸受剛性は後者の方が高い。 このため、スイングパッド軸受の方が安定した軸受性能が得られる。

## 参考文献

工業基礎振動学 齊藤秀雄

養賢堂

弾性論 金多 潔 荒川宗夫  
坂口 昇 森 哲朗

コロナ社

すべり軸受の静特性および動特性資料集  
日本機械学会編

日本工業出版

---

## 謝辞

この修士論文の完成にあたり、スイングパッド軸受の製作に御助力いただいたイーグル工業株式会社の田上氏、塩見氏に感謝申し上げます。

そして、多岐に渡り終始熱心なご指導と懇切丁寧にご教示をいただいた岩本勝美助教授に厚くお礼申し上げます。

また、装置の設計や加工だけでなく様々な方面にわたり、多くの御協力と御助言を頂きました、大石誠治技官、外山一雄技官には深く感謝いたします。 また、実験や論文作成などに際し多大に協力してくれた修士一年の香高和男君に深くお礼を申し上げます。

## 付録1 各計測・計算に使用したプログラム

### 1-1 AD変換ボードより取り込んだデータを フロッピーディスクに記録するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include <graph.c>

#define inp      inportb
#define inpw     inport
#define outp     outportb
#define outpw    outport
#define IOPORT 0x12d0

adt(int a)
{
    unsigned int j;
    outp(IOPORT,a);          /*チャンネル指定          */
    inp(IOPORT);             /*EOCを 0 (Low) にする    */
    while((inp(IOPORT+2) & 0x80)!=0); /*EOC 待ち (MSB が 1 になるのを待つ)*/
    j = inpw(IOPORT) & 0xffff; /*変換値を読み、        */
    return j;
}

main()
{
    char file[64], st;
    unsigned int i, a, b, x[4098], y[4098];
    FILE *fp;
    startgraph();
    clrscr();
    cleardevice();

    y[0]=0xff;
    x[0]=0xff;

    printf("      Input Any Kye !");
    scanf("%s",&st);

    for(i=1;i<4097;i++)      /*データを取り込む*/
    {
        y[i]=adt(6);
        x[i]=adt(7);
    }

    for(i=1;i<4097;i++)      /*データの画像表示*/
    {
        line(x[i]/13,y[i]/13,x[i-1]/13,y[i-1]/13);
    }
}
```

---

```
printf(" Input Data File-name : "); /*ファイルにセーブ*/
scanf("%s", file);
printf("\n");

if(NULL == (fp = fopen(file, "w")))
{
    printf("\n Cannot Open File : %s\n",file);
    exit(1);
}
for(i=1;i<4097;i++)
{
    fprintf(fp, " %x",y[i]);
    fprintf(fp, " %x",x[i]);
}
fclose(fp);
closegraph();
}
```

---

## 1 - 2 AD 変換からのデータより軸芯の軌跡に直して リアルタイムに表示するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include <graph.c>

#define inp      inportb
#define inpw     inport
#define outp     outportb
#define outpw    outport
#define IOPORT 0x12d0

main()
{
    int i, j, d, datx, daty, a;
    startgraph();
    clrscr();
    cleardevice();
    a=1;
    for(;;)
    {
        inp(IOPORT);          /*EOCを 0 (Low) にする      */
        outp(IOPORT,6);       /*チャンネル指定によって変換が開始する*/
        while((d=inp(IOPORT)) > 0); /*変換終了待ち (BSY が 1 になるのを待つ)*/
        datx = d & 0xff;      /*変換値を読み、          */

        inp(IOPORT);          /*EOCを 0 (Low) にする      */
        outp(IOPORT,7);       /*チャンネル指定によって変換が開始する*/
        while((d=inp(IOPORT)) > 0); /*変換終了待ち (BSY が 1 になるのを待つ)*/
        daty = d & 0xff;      /*変換値を読み、          */

        setcolor(a);
        circle(datx/10,daty/10,3);
        printf("%d %d  ¥n",datx,daty);
        a++;
        if(a == 15)
        {
            a=1;
        }
    }
    closegraph();
}
```

### 1 - 3 軸芯のふれ半径を計測するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include <graph.h>
#include <math.h>

#define inp      inportb
#define inpw     inport
#define outp     outportb
#define outpw    outport
#define IOPORT 0x12d0

adt(int a)
{
    unsigned int j;
    outp(IOPORT,a);          /*チャンネル指定          */
    inp(IOPORT);              /*EOCを 0 (Low) にする      */
    while((inp(IOPORT+2) & 0x80)==0); /*EOC 待ち (MSB が 1 になるのを待つ)*/
    j = inpw(IOPORT) & 0xffff; /*変換値を読み、          */
    return j;
}

main()
{
    int d;
    unsigned int a, b, c, i, j, tx1, ty1, tx2, ty2, by;
    unsigned int nx1[2001], ny1[2001], nx2[2001], ny2[2001];
    double avgx1, avgx2, avgy1, avgy2, mr1, mr2, tr1, tr2, ch3, ch5, ch6, ch7;

    startgraph();
    clrscr();
    cleardevice();

    setcolor(6);
    by=13;
    ch3=.0035163817;
    ch5=.00099607115;
    ch6=.0008580564;
    ch7=.001442431;
    avgx1=avgx2=avgy1=avgy2=0;
    mr1=mr2=0;
    rectangle(0,0,319,399);
    rectangle(320,0,639,399);

    for(i=0;i<2000;i++)
    {
        nx1[i]=0;
        nx2[i]=0;
        ny1[i]=0;
        ny2[i]=0;
    }
}
```



```
    }

    tx1 = nx1[1] = adt(6);
    ty1 = ny1[1] = adt(7);

    /*
    tx2 = nx2[1] = adt(3);
    ty2 = ny2[1] = adt(5);
    */

    for(i=2;i<2001;i++)
    {
        nx1[i] = adt(6);
        ny1[i] = adt(7);
        /*
        nx2[i] = adt(3);
        ny2[i] = adt(5);
        */
        setcolor(2);
        line(tx1/by,ty1/by+50,nx1[i]/by,ny1[i]/by+50);
    /*
        setcolor(4);
        line(tx2/by+320,ty2/by+50,nx2[i]/by+320,ny2[i]/by+50);
        */
        tx1=nx1[i];
        ty1=ny1[i];
        tx2=nx2[i];
        ty2=ny2[i];
    }

    for(i=1;i<2001;i++)
    {
        avgx1=avgx1+((double)nx1[i])/2000;
        avgy1=avgy1+((double)ny1[i])/2000;
        avgx2=avgx2+((double)nx2[i])/2000;
        avgy2=avgy2+((double)ny2[i])/2000;
    }

    for(i=1;i<2001;i++)
    {
        tr1=sqrt(((double)nx1[i]-avgx1)*((double)nx1[i]-avgx1)*ch6*ch6+((double)ny1[i]-avgy1)*
        ((double)ny1[i]-avgy1)*ch7*ch7);
        tr2=sqrt(((double)nx2[i]-avgx2)*((double)nx2[i]-avgx2)*ch3*ch3+((double)ny2[i]-avgy2)*
        ((double)ny2[i]-avgy2)*ch5*ch5);

        if(mr1<tr1)
        {
            mr1=tr1;
        }
        if(mr2<tr2)
        {
            mr2=tr2;
        }
    }
```

---

```
printf("最大振幅は 6, 7 チャンネルで%3.5lf [mm] 3, 5 チャンネルで%3.5lf [mm] ",mr1,mr2);  
delay(2000);  
closegraph();  
}
```

---

## 1 - 4 軸直径より各長さの固有振動数を計算するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main()
{
    int rpm, i;
    float d, e, g11, g12, g13, g21, g22, g23, g31, g32, g33, j, pi, y1, y2, y3;
    float x, w1, w2, w3, omg, omg2, l1, l2, l3, l4, l;
    e=21000;
    pi=3.141592;
    printf("軸直径を入力してください。(mm)");
    scanf("%f",&d);
    j=pi*d*d*d*d/64;
    w2=1.891;
    for(i=0;i<50;i++)
    {
        l=(float)(119+5*i);
        x=(297-l)/18;
        l1=l4=33-x;
        l2=l3=x+l+28;
        w1=pi*7.8*(.5*.5*l/10+1.5*1.5*6.6+1.2*1.2)/1000;
        w3=pi*7.8*(.5*.5*l/10+1.6*1.6*4.1+1.7*1.7*2.5+1.35*1.35*3)/1000;
        g33=g11=(l1*l1*(l2+l3+l4)*(l2+l3+l4))/(3*(l1+l2+l3+l4)*e*j);
        g22=((l1+l2+l3+l4)*(l1+l2+l3+l4))/(48*e*j);
        g21=g23=g32=g12=(l1*(3*(l1+l2+l3+l4)*(l1+l2+l3+l4)-4*l1*l1))/(48*e*j);
        g31=g13=(l1*l4*((l1+l2+l3)*(l1+l2+l3)+2*l4*(l1+l2+l3)-l1*l1))/(6*(l1+l2+l3+l4)*e*j);
        y1=g11*w1+g12*w2+g13*w3;
        y2=g21*w1+g22*w2+g23*w3;
        y3=g31*w1+g32*w2+g33*w3;
        printf("\n %f %f %f ",y1,y2,y3);
        omg2=((w1*y1+w2*y2+w3*y3)/(w1*y1*y1+w2*y2*y2+w3*y3*y3))*9800;
        omg=(float)(pow(((double)(omg2)),(.5)));
        rpm=(int)(omg*30/pi);
        printf("\n軸の長さが%dcmの時,回転数は%d r p mです。",(int)(l1+l2+l3+l4+106+.5),rpm);
    }
}
```

## 1 - 5 軸の直径より各長さのたわみを計算するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

main()
{
    int j, ji;
    float d, w, l, i, rtw, del, pi=3.1415926, e=2100000;
    l=36;
    printf("Yn軸の代表直径を入力してください (c m) ");
    scanf("%f",&d);
    i=pi*d*d*d*d/64;
    for(ji=0;ji<15;ji++)
    {
        l=l+2;
        w=1.8/l;
        printf("Yn軸の長さが%3fの時の. . . ",l);
        for(j=0;j<10;j++)
        {
            rtw=(float)(j);
            del=w*l*l*l/(384*e*i)+rtw*l*l/(192*e*i);
            printf("Ynロータ重量%f k g の、たわみは%fm m です。",(rtw+w*l),del);
        }
    }
}
```

## 1 - 6 ギャプセンサのキャリブレーション用のプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include <graph.c>
#include <time.h>
#include <process.h>

#define inp      inportb
#define inpw     inport
#define outp     outportb
#define outpw    outport
#define IOPORT 0x12d0

adc(
    int channel, /*[I] チャンネル番号 0 - 7 */
    int wait_num
)
{
    int i, j, d, dat;

    inp(IOPORT); /*EOCを 0 (Low) にする */
    outp(IOPORT,channel); /*チャンネル指定によって変換が開始する*/
    while((d=inpw(IOPORT)) > 0); /*変換終了待ち (BSY が 1 になるのを待つ)*/
    dat = d & 0xffff; /*変換値を読み、 */
    /*上位 4 ビットマスク後、
    *格納 */
    for(j=0; j<wait_num; j++); /*ソフトによるデータ取り込みタイミング*/
    return (dat);
}

main()
{
    char pro, wthr;
    char file[64];
    int tmp, oltmp, wet, lbd, wgh, ch, times, cnt, minit, sec;
    unsigned int a, b, a1, b1, i, k;
    double leng;
    FILE *fp;
    k=2;
    times=6;
    cnt=16;
    minit=6;
    sec=5;
    startgraph();
    clrscr();
    cleardevice();
    pro='a';
    printf("マイクロメータの読みの倍率を入力 (m a x を 6 3 9 + a にするように) ");
    scanf("%d",&minit);
    printf("マイクロメータの読みの補正を入力 (- a ) ");
```

```
scanf("%d",&sec);
while(pro != 'x' && pro != 'X'){
    printf("\nData input or Graph & data out or End ? ");
    if(pro != 'x' && pro != 'X' && pro != 'd' && pro != 'D' && pro != 'g' && pro != 'G' && pro != 'E' &&
    pro != 'e')
    {
        scanf("%s",&pro);
    }
    if(pro == 'd' || pro == 'D')
    {
        printf(" Input Data File-name : ");
        scanf("%s", file);
        printf("\nInput channel :");
        scanf("%d",&ch);
        printf("\n");

        if(NULL == (fp = fopen(file, "w")))
        {
            printf("\n Cannot Open File : %s\n",file);
            exit(1);
        }
        printf("\n %d ",1);
        scanf("%lf",&leng);
        fprintf(fp, " %f",leng);
        a=adc(ch,100);
        fprintf(fp, " %x",a);
        printf(" %x ",a);
        a1=a/times-cnt;
        b1=(int)(leng/(double)minit-(double)sec);
        for(i=0;i<49;++i)
        {
            printf("\n %d ",i+2);
            scanf("%lf",&leng);
            fprintf(fp, " %f",leng);
            a=adc(ch,100);
            fprintf(fp, " %x",a);
            printf(" %x ",a);
        }
        a=a/times-cnt;
        b=(int)(leng/minit-sec);
        line(a1,b1,a,b);
        a1=a;
        b1=b;
    }

    fclose(fp);
}

else if(pro == 'g' || pro == 'G')
{

    printf(" Input Dump File-name : ");
    scanf("%s", file);
    printf("\n");

    if(NULL == (fp = fopen(file, "r")))
```

```
{
printf("¥7¥n Cannot Open File : %s¥n¥n",file);
exit(1);
}

clrscr();
cleardevice();*/
setcolor(4);
setlinestyle(SOLID_LINE,0,THICK_WIDTH);
rectangle(9,9,629,389);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
setcolor(14);

fscanf(fp,"%lf",&leng);
fscanf(fp,"%x",&a);
printf(" %d %3.3f¥n",a,leng);
a1=a/times-cnt;
b1=(int)(leng/minit+sec);
setcolor(k);

for(i=0;i<49;++i)
{
fscanf(fp,"%lf",&leng);
fscanf(fp,"%x",&a);
printf(" %d %d¥n",a1,b1);
a=a/times-cnt;
b=(int)(leng/minit-sec);
line(a1,b1,a,b);
a1=a;
b1=b;
}

fclose(fp);
k++;
}
if(pro == 'e' || pro == 'E')
{
exit(0);
}
pro='a';
}

/*
closegraph();
*/
}
```

## 1 - 7 AD 変換ボードより取り入れたフロッピディスクのデータをグラフィック画面に表示するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>
#include <graph.c>

#define inp      inportb
#define inpw     inport
#define outp     outportb
#define outpw    outport
#define IOPORT 0x12d0

main()
{
    char file[64];
    unsigned int i, a, b, a1, b1, by;
    FILE *fp;
    startgraph();
    clrscr();
    cleardevice();
    by=13;

    printf(" Input Dump File-name : ");
    scanf("%s", file);
    printf("\n");

    if(NULL == (fp = fopen(file, "r")))
    {
        printf("\n Cannot Open File : %s\n",file);
        exit(1);
    }

    setcolor(4);
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    rectangle(1,1,638,398);
    line(319,1,319,398);
    setlinestyle(SOLID_LINE,0,NORM_WIDTH);

    fscanf(fp,"%x",&a);
    a1=a/by;
    fscanf(fp,"%x",&b);
    b1=b/by;
    printf(" %d %d\n",a1,b1);
    for(i=0;i<4094;i++)
    {
        fscanf(fp,"%x",&a);
        fscanf(fp,"%x",&b);
        printf(" %d %d\n",a/by,b/by);
        setcolor(5);
        line(a1,b1,a/by,b/by);
    }
}
```



---

```
    a1=a/by;  
    b1=b/by;  
}
```

```
fclose(fp);  
closegraph();  
}
```

---

## 1 - 8 任意の回転数一回分の FFT 計算をして グラフィック表示するプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <alloc.h>
#include <graphics.h>
#include <graph.c>

#define l 12
#define flg -1
#define dat 4096
#define mex 40
#define mey 34
#define hor 4
#define var 1.4

int xa[4098],ya[4098];
double far *x, *y;

main()
{
    char g;
    int i, j, k, n, m;
    startgraph();

    clrscr();
    cleardevice();

    x=farmalloc((dat+2)*sizeof(double));
    y=farmalloc((dat+2)*sizeof(double));

    initscr();
    setcolor(0);
    outtextxy(200,10,"水平方向");

    read_dt();
    for(m=0;m<dat;m++)
    {
        x[m+1]=(double)ya[m];
        y[m+1]=0;
    }
    fft_fn();
    out_dt(0,0);
}

read_dt()
{
    unsigned int i, j, a, b;
    char bc[2], cc[2], file[64]="j151.dat";
    FILE *fp;
```

```

printf(" 変換するファイル名を入力してください。:");
scanf("%s", file);
printf("%n");

if(NULL == (fp = fopen(file, "r")))
{
    printf("%7Yn ファイルがおまへん。      : %sYnYn",file);
    exit(1);
}

for(i=0;i<dat;i++)
{
    fscanf(fp,"%x",&a);
    xa[i]=a;

    fscanf(fp,"%x",&b);
    ya[i]=b;
}
fclose(fp);
}

out_dt(int c,int co)
{
    int i, x1, y1, x2, y2;
    double ps;

    clrscr();
    x1=50+co*3;
    y1=300-co*4;

    setcolor(c);
    setfillstyle(SOLID_FILL,0);

    for(i=1;i<(int)(dat/40);i++)
    {
        ps=4.343*log(x[i]*x[i]+y[i]*y[i]);

        x2=50+i*hor+co*3;
        y2=300-(int)(ps*var+.5)-co*4;

        setcolor(c);
        line(x2,y2,x1,y1);
        x1=x2;
        y1=y2;
    }
}

fft_fn()
{
    int i, n, nv2, nm1, lmx, lm, lix, l0, l1, j, j1, j2, k;
    double scl, arg, c, s, t1, t2;
    n=(int)pow(2,l);
    lmx=n;
    scl=6.283185303/((double)n);

```

```
for(l0=1;l0<=l;l0++)
{
    lix=lmx;
    lmx=lmx/2;
    arg=0;
    for(lm=1;lm<=lmx;lm++)
    {
        c=cos(arg);
        s=sin(fl*arg);
        arg=arg+scl;
        for(l1=lix;l1<=n;l1=l1+lix)
        {
            j1=l1-lix+lm;
            j2=j1+lmx;
            t1=x[j1]-x[j2];
            t2=y[j1]-y[j2];
            x[j1]=x[j1]+x[j2];
            y[j1]=y[j1]+y[j2];
            x[j2]=c*t1+s*t2;
            y[j2]=c*t2-s*t1;
        }
    }
    scl=2*scl;
}
nv2=n/2;
nm1=n--;
j=1;
for(i=1;i<=nm1;i++)
{
    if((i-j) < 0)
    {
        t1=x[j];
        t2=y[j];
        x[j]=x[i];
        y[j]=y[i];
        x[i]=t1;
        y[i]=t2;
    }
    k=nv2;
    while((k-j) < 0)
    {
        j=j-k;
        k=k/2;
        if(k==0)
        {
            break;
        }
    }
    j=j+k;
}
if(fl*arg < 0)
{
    for(i=1;i<=n;i++)
    {
        x[i]=x[i]/(double)n;
    }
}
```

---

```
        y[i]=y[i]/(double)n;
    }
}

initscr()
{
char a[5];
int f, i;

setfillstyle(SOLID_FILL,15);
bar(0,0,639,399);
setcolor(0);
line(50,50,50,350);
line(50,300,600,300);
f=10;

for(i=0;i<13;i++)
{
    line(50+i*mex,295,50+i*mex,305);
    itoa(((int)(i*12.20703125)),a,10);
    outtextxy(50+i*mex,310,a);
    f=f+mex;
}
outtextxy(562,310,"(Hz)");

for(i=0;i<9;i++)
{
    line(45,62+i*mey,55,62+i*mey);
    itoa((7-i)*10,a,10);
    setcolor(0);
    outtextxy(20,62+i*mey,a);
}
outtextxy(10,75,"(dB)");
}
```

---

# 1 - 9 連続した回転 1 パラメータ分のデータの FFT 計算をして疑似的な 3 次元表示のグラフィック をするプログラム

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <alloc.h>
#include <graphics.h>
#include <graph.c>

#define l 12
#define flg -1
#define dat 4096
#define mex 40
#define mey 34
#define hor 4
#define var 1.4

int xa[4098],ya[4098];
double far *x, *y;

main()
{
char g;
int i, j, k, n, m;
startgraph();

clrscr();
cleardevice();

x=farmalloc((dat+2)*sizeof(double));
y=farmalloc((dat+2)*sizeof(double));

printf("最初の回転数を入力してください。 : ");
scanf("%d",&i);
printf("最後の回転数を入力してください。 : ");
scanf("%d",&j);
i=i/100;
j=j/100;

k=j-i;

/*
initscr();
read_dt();

for(m=0;m<dat;m++)
{
x[m+1]=(double)xa[m];
y[m+1]=0;
}
```

```

fft_fn();

out_dt(3);
setcolor(13);
outtextxy(200,10,"垂直方向");
*/

initscr(i);
setcolor(12);
outtextxy(200,10,"水平方向");

for(n=j;n>(i-1);n--)
{
    read_dt(n);
    for(m=0;m<dat;m++)
    {
        x[m+1]=(double)ya[m];
        y[m+1]=0;
    }
    fft_fn();
    out_dt(4,k);
    k--;
}
initscr(i);
}

read_dt(int num)
{
    unsigned int i, j, a, b;
    char bc[2], cc[2], file[64]="j151.dat";
    FILE *fp;

    /*
    printf(" 変換するファイル名を入力してください。:");
    scanf("%s", file);
    printf("\n");
    */

    j=num/10;
    itoa(j,bc,10);
    file[1]=bc[0];
    j=num-j*10;
    itoa(j,cc,10);
    file[2]=cc[0];

    if(NULL == (fp = fopen(file, "r")))
    {
        printf("\n7 ファイルがおまへん。      : %s\n",file);
        exit(1);
    }

    for(i=0;i<dat;i++)
    {

```

```
fscanf(fp,"%x",&a);
xa[i]=a;

fscanf(fp,"%x",&b);
ya[i]=b;
}
fclose(fp);
}

out_dt(int c,int co)
{
int i, x1, y1, x2, y2, pol[12];
double ps;

clrscr();
x1=50+co*3;
y1=300-co*4;

setcolor(c);
setfillstyle(SOLID_FILL,0);
pol[3]=399;
pol[5]=399;

for(i=1;i<(int)(dat/40);i++)
{
ps=4.343*log(x[i]*x[i]+y[i]*y[i]);
/*
fprintf(stdprn," %2.5lf %2.5lf %n",x[i],y[i]);
printf(" %4d %2.5lf %n",i,ps);
*/

x2=50+i*hor+co*3;
y2=300-(int)(ps*var+.5)-co*4;

setcolor(0);
pol[0]=x1;
pol[1]=y1;
pol[2]=x1;
pol[4]=x2;
pol[6]=x2;
pol[7]=y2;
pol[8]=x1;
pol[9]=y1;
fillpoly(4,pol);

setcolor(c);
line(x2,y2,x1,y1);
x1=x2+1;
y1=y2;
}
}

fft_fn()
{
int i, n, nv2, nm1, lmx, lm, lix, l0, l1, j, j1, j2, k;
```



```
double scl, arg, c, s, t1, t2;
n=(int)pow(2,l);
lmx=n;
scl=6.283185303/((double)n);
for(l0=1;l0<=l;l0++)
{
    lix=lmx;
    lmx=lmx/2;
    arg=0;
    for(lm=1;lm<=lmx;lm++)
    {
        c=cos(arg);
        s=sin(flq*arg);
        arg=arg+scl;
        for(l1=lix;l1<=n;l1=l1+lix)
        {
            j1=l1-lix+lm;
            j2=j1+lmx;
            t1=x[j1]-x[j2];
            t2=y[j1]-y[j2];
            x[j1]=x[j1]+x[j2];
            y[j1]=y[j1]+y[j2];
            x[j2]=c*t1+s*t2;
            y[j2]=c*t2-s*t1;
        }
        scl=2*scl;
    }
nv2=n/2;
nm1=n--;
j=1;
for(i=1;i<=nm1;i++)
{
    if((i-j) < 0)
    {
        t1=x[j];
        t2=y[j];
        x[j]=x[i];
        y[j]=y[i];
        x[i]=t1;
        y[i]=t2;
    }
    k=nv2;
    while((k-j) < 0)
    {
        j=j-k;
        k=k/2;
        if(k==0)
        {
            break;
        }
    }
    j=j+k;
}
if(flq < 0)
```

```
{
    for(i=1;i<=n;i++)
    {
        x[i]=x[i]/(double)n;
        y[i]=y[i]/(double)n;
    }
}

initscr(int j)
{
    char a[5];
    int f, i;

    setcolor(2);
    line(50,50,50,350);
    line(50,300,600,300);
    line(23,336,185,120);
    f=10;

    itoa(j*100,a,10);
    setcolor(14);
    outtextxy(8,285,a);
    for(i=1;i<5;i++)
    {
        setcolor(6);
        line(45+30*i,300-40*i,55+30*i,300-40*i);
        itoa(j*100+i*1000,a,10);
        setcolor(14);
        outtextxy(5+30*i,285-40*i,a);
    }
    outtextxy(10,312,"(rpm)");

    for(i=0;i<13;i++)
    {
        setcolor(6);
        line(50+i*mex,295,50+i*mex,305);
        itoa(((int)(i*12.20703125)),a,10);
        setcolor(14);
        outtextxy(50+i*mex,310,a);
        f=f+mex;
    }
    outtextxy(562,310,"(Hz)");

    for(i=0;i<9;i++)
    {
        setcolor(6);
        line(45,62+i*mey,55,62+i*mey);
        itoa((7-i)*10,a,10);
        setcolor(14);
        outtextxy(20,62+i*mey,a);
    }
    outtextxy(10,75,"(dB)");
}
```

# 1 - 1 1 油膜の圧力と、パッドとゴムの弾性変形の 有限要素法計算による軸に掛かる力の計算

/\*\*\*\*\*\*軸受 3 0 mm V e r \*\*\*\*\*\*/

#include <stdio.h>

```
#define pi          3.1415926 /*           $\pi$           */
#define r1          15.00    /*          軸の半径          */
#define meshx1  10    /*          軸方向メッシュ / 2          */
#define meshy1   8     /*          半径方向のメッシュ          */
#define mat        90     /*          ローカル変数の配列の数          */
#define rpm        5000    /*          回転速度          */
#define mu         7.6     /*          油の粘度          */
#define n1         80     /*          圧力の計算回数          */
#define r2         15.23   /*          軸受の半径          */
#define meshx2     10     /*          ゴムの半径方向メッシュ          */
#define meshy2     8     /*          ゴムの軸回転方向メッシュ          */
#define E1         15     /*          ブロンズのヤング率          */
#define Nu1        .47    /*          ブリズンのポアソン比          */
#define E2         .00000063 /*          ゴムのヤング率          */
#define Nu2        0.34   /*          ゴムのポアソン比          */
#define n2         80     /*          油膜厚さの計算回数          */
#define t          30     /*          軸受長さ          */
```

```
/*
    10mm    12mm    12mm
jiku 30.00,    30.06,    29.95
    30mm
uke 30.45,    30.27,    30.21
*/
```

double Ka[90][90],p[90],x[90],pr[10],hpd[10];

main()

```
{
int i, j;
double determ, Ep, adv, ps1, ps2, ang;
double intgforp(), calforang();
unsigned double dps;
Ep=0.0;
ps1=0;
dps=1;
```

```
while(Ep < 0.23)
{
    for(i=0;i < 360;i++)
    {
        pxinit();
        while(dps < 0.0000001)
        {
            hpressk( Ep, (double)i);
            determ = invmat(1);
            matcal(1);
```

```
press();

dhmovek();
determ = invmat(2);
matcal(2);
intgfor();
ps2=intgforp();
ang=calforang();
if(ps1 == 0)
{
break;
}
else
{
dps=fabs((ps1+ps2)/ps1);
}
ps1=ps2;
}
/*
recfor(Ep,i,ps2,ang);
*/
}
Ep=Ep+.01
}

}

ang=calforang()
{
int i;
double sumx, sumy, ang;

sumx=sumy=0;

for(i=0;i<9;i++)
{
sumy=sumy+pr[i]*sin(i*10);
sumx=sumx+pr[i]*cos(i*10);
}
ang=atan(sumy/sumx);
return (ang);
}

double intgforp()
{
double sum;

s=0;

for(i=1;i<4;i++)
{
sum=sum+2*pi*r2*(2*pr[i*2]+p[i*2+1])/27;
}
return (sum);
}
```

```
dhcal()
{
int i;
for(i=1;i<9;i++)
{
dph[i]=x[i*10+1];
}
}

recfor()
{
int i, j, roop;
double sum;
roop=meshx1/2+1;
for(i=1;i<9;i++)
{
sum=0;
for(j=1;j<roop;j++)
{
sum=sum+x[meshx1*(i-1)+j*2-1]*4+x[meshx1*(i-1)+j*2]*2;
}
pr[i]=sum+x[meshx1*i-1];
}
}

hcal(int i,double Ep,double adv)
{
double a;
a=1+Ep*cos(i*10+adv)+dph[i];
return (a);
}

matcal(int sw)
{
int i, j, n;

n=condn(sw);

for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
x[i]=x[i]+p[j]*Ka[i][j];
}
}
}

condn(int sw)
{
int n;
if(sw == 1)
{
n=n1;
}
else
```

```
{
    n=n2;
}
return (n);
}

invmat(int sw)
{
double fabs();
int ipivot[170],index[170][170];
int l,m,n,jrow,jcolumn,i,irow,icolumn,j,k;
double tm,swap,determ,amax,pivot[170];
determ=1;

n=cond(sw);

for(j=0;j<n;++j)
{
    ipivot[j]=0;
}

for(i=0;i<n;++i)
{
    amax=0.0;
    for(j=0;j<n;++j)
    {
        if(ipivot[j] != 1)
        {
            for(k=0;k<n;++k)
            {
                if( (ipivot[k]-1) > 0 )
                {
                    return(determ);
                }
                if(((ipivot[k]-1) < 0 ) && ((fabs(amax)-fabs(Ka[j][k])) < 0 ))
                {
                    irow=j;
                    icolumn=k;
                    amax=Ka[j][k];
                }
            }
        }
    }
    ipivot[icolumn]=ipivot[icolumn]+1;
    if( (irow-icolumn) != 0 )
    {
        determ = -determ;
        for(l=0;l<n;++l)
        {
            swap=Ka[irow][l];
            Ka[irow][l]=Ka[icolumn][l];
            Ka[icolumn][l]=swap;
        }
    }
    index[i][0]=irow;
}
```

```
index[i][1]=icolum;
pivot[i]=Ka[icolum][icolum];
determ = determ*pivot[i];
Ka[icolum][icolum]=1.0;
for(l=0;l<n;++l)
{
    Ka[icolum][l]=Ka[icolum][l]/pivot[i];
}
for(m=0;m<n;++m)
{
    if( (m-icolum) != 0 )
    {
        tm=Ka[m][icolum];
        Ka[m][icolum]=0.0 ;
        for(l=0;l<n;++l)
        {
            Ka[m][l]=Ka[m][l]-Ka[icolum][l]*tm;
        }
    }
}
}
for(i=0;i<n;++i)
{
    l=n+1-i;
    if( (index[l][0]-index[l][1]) != 0 )
    {
        jrow=index[l][0];
        jcolum=index[l][1];
        for(k=1;k<=n;++k)
        {
            swap=Ka[k][jrow];
            Ka[k][jrow]=Ka[k][jcolum];
            Ka[k][jcolum]=swap;
        }
    }
}
return(determ);
}

double fabs(x)
double x;
{
    if (x < 0.0)
    {
        x = -x;
    }
    return(x);
}

pxinit()
{
    int i, j;
    for(i=0;i<mat;i++)
    {
```

```
p[i]=0;
x[i]=0;
for(j=0;j<mat;j++)
{
    Ka[i][j]=0;
}
}
for(i=0;i<meshy1+1;i++)
{
    pr[i]=0;
    hpd[i]=0;
}
}

hpressk(double Ep,double adv)
{
    int i, j, k, l, m, ii, ji, ki, ij, jj, kj;
    double a, y, x, sq1, em, ux, H3;
    double kk[8][8], hcal(int,double);

    q1=q2=q3=0;
    ux=2*rpm*pi*r1/60;
    a=2*pi*r1/36;
    sq1=1.5*a/2;

    for(i=0;i<n1;i++)
    {
        for(j=0;j<n1;j++)
        {
            Ka[i][j]=0;
        }
    }

    for(i=0;i<meshy1-1;i++)
    {
        for(j=0;j<meshx1-1;j++)
        {
            ii=i*meshx1+j; ij=i*meshx1+j+1;
            ji=i*meshx1+j+1; kj=(i+1)*meshx1+j;
            ki=(i+1)*meshx1+j; jj=(i+1)*meshx1+j+1;

            h1=hcal(j,Ep,adv);
            h2=hcal(j+1,Ep,adv);
            h3=hcal(j,Ep,adv);

            H3=(h1*h1+h2*h2+h3*h3)*(h1+h2+h3)+(h1*h2*h3);

            p[ii]=p[ii]+(h1+h2+h3)*ux/4KB+1.5*a*h1+1.5*a*q1;
            p[ij]=p[ij]+1.5*a*h2+1.5*a*q2;
            p[ik]=p[ik]+(h1+h2+h3)*ux*(-1)/4+1.5*a*h3+1.5*a*q3;

            for(k=0;k<7;k++)
            {
                for(l=0;l<7;l++)
                {
```



---

```

        kk[k][l]=0;
    }
}

kk[0][0]=H3*(1.5*1.5+a*a)/240*mu;
kk[1][1]=H3*a*a/240*mu;
kk[1][0]=kk[0][1]=H3*a*a*(-1)/240*mu;
kk[0][2]=kk[2][0]=H3*1.5*1.5*(-1)/240*mu;
kk[1][2]=kk[2][1]=0;
kk[2][2]=H3*1.5*1.5/240*mu;

Ka[2*ii][2*ii]=Ka[2*ii][2*ii]+kk[0][0];
Ka[2*ii][2*ii+1]=Ka[2*ii][2*ii+1]+kk[0][1];
Ka[2*ii][2*ji]=Ka[2*ii][2*ji]+kk[0][2];
Ka[2*ii][2*ji+1]=Ka[2*ii][2*ji+1]+kk[0][3];
Ka[2*ii][2*ki]=Ka[2*ii][2*ki]+kk[0][4];
Ka[2*ii][2*ki+1]=Ka[2*ii][2*ki+1]+kk[0][5];

Ka[2*ii+1][2*ii]=Ka[2*ii+1][2*ii]+kk[1][0];
Ka[2*ii+1][2*ii+1]=Ka[2*ii+1][2*ii+1]+kk[1][1];
Ka[2*ii+1][2*ji]=Ka[2*ii+1][2*ji]+kk[1][2];
Ka[2*ii+1][2*ji+1]=Ka[2*ii+1][2*ji+1]+kk[1][3];
Ka[2*ii+1][2*ki]=Ka[2*ii+1][2*ki]+kk[1][4];
Ka[2*ii+1][2*ki+1]=Ka[2*ii+1][2*ki+1]+kk[1][5];

Ka[2*ji][2*ii]=Ka[2*ji][2*ii]+kk[2][0];
Ka[2*ji][2*ii+1]=Ka[2*ji][2*ii+1]+kk[2][1];
Ka[2*ji][2*ji]=Ka[2*ji][2*ji]+kk[2][2];
Ka[2*ji][2*ji+1]=Ka[2*ji][2*ji+1]+kk[2][3];
Ka[2*ji][2*ki]=Ka[2*ji][2*ki]+kk[2][4];
Ka[2*ji][2*ki+1]=Ka[2*ji][2*ki+1]+kk[2][5];

Ka[2*ji+1][2*ii]=Ka[2*ji+1][2*ii]+kk[3][0];
Ka[2*ji+1][2*ii+1]=Ka[2*ji+1][2*ii+1]+kk[3][1];
Ka[2*ji+1][2*ji]=Ka[2*ji+1][2*ji]+kk[3][2];
Ka[2*ji+1][2*ji+1]=Ka[2*ji+1][2*ji+1]+kk[3][3];
Ka[2*ji+1][2*ki]=Ka[2*ji+1][2*ki]+kk[3][4];
Ka[2*ji+1][2*ki+1]=Ka[2*ji+1][2*ki+1]+kk[3][5];

Ka[2*ki][2*ii]=Ka[2*ki][2*ii]+kk[4][0];
Ka[2*ki][2*ii+1]=Ka[2*ki][2*ii+1]+kk[4][1];
Ka[2*ki][2*ji]=Ka[2*ki][2*ji]+kk[4][2];
Ka[2*ki][2*ji+1]=Ka[2*ki][2*ji+1]+kk[4][3];
Ka[2*ki][2*ki]=Ka[2*ki][2*ki]+kk[4][4];
Ka[2*ki][2*ki+1]=Ka[2*ki][2*ki+1]+kk[4][5];

Ka[2*ki+1][2*ii]=Ka[2*ki+1][2*ii]+kk[5][0];
Ka[2*ki+1][2*ii+1]=Ka[2*ki+1][2*ii+1]+kk[5][1];
Ka[2*ki+1][2*ji]=Ka[2*ki+1][2*ji]+kk[5][2];
Ka[2*ki+1][2*ji+1]=Ka[2*ki+1][2*ji+1]+kk[5][3];
Ka[2*ki+1][2*ki]=Ka[2*ki+1][2*ki]+kk[5][4];
Ka[2*ki+1][2*ki+1]=Ka[2*ki+1][2*ki+1]+kk[5][5];

h1=hcal(j+1,Ep,adv);
h2=hcal(j+1,Ep,adv);

```

---

---

```

h3=hcal(j,Ep,adv);

H3=(h1*h1+h2*h2+h3*h3)*(h1+h2+h3)+(h1*h2*h3);

p[ii]=p[ii]+(h1+h2+h3)*ux/4KB+1.5*a*h1+1.5*a*q1;
p[ij]=p[ij]+1.5*a*h2+1.5*a*q2;
p[ik]=p[ik]+(h1+h2+h3)*ux*(-1)/4+1.5*a*h3+1.5*a*q3;

for(k=0;k<7;k++)
{
for(l=0;l<7;l++)
{
kk[k][l]=0;
}
}

kk[0][0]=H3*(1.5*1.5+a*a)/240*mu;
kk[1][1]=H3*a*a/240*mu;
kk[1][0]=kk[0][1]=H3*a*a*(-1)/240*mu;
kk[0][2]=kk[2][0]=H3*1.5*1.5*(-1)/240*mu;
kk[1][2]=kk[2][1]=0;
kk[2][2]=H3*1.5*1.5/240*mu;

Ka[2*ij][2*ij]=Ka[2*ij][2*ij]+kk[0][0];
Ka[2*ij][2*ij+1]=Ka[2*ij][2*ij+1]+kk[0][1];
Ka[2*ij][2*jj]=Ka[2*ij][2*jj]+kk[0][2];
Ka[2*ij][2*jj+1]=Ka[2*ij][2*jj+1]+kk[0][3];
Ka[2*ij][2*kj]=Ka[2*ij][2*kj]+kk[0][4];
Ka[2*ij][2*kj+1]=Ka[2*ij][2*kj+1]+kk[0][5];

Ka[2*ij+1][2*ij]=Ka[2*ij+1][2*ij]+kk[1][0];
Ka[2*ij+1][2*ij+1]=Ka[2*ij+1][2*ij+1]+kk[1][1];
Ka[2*ij+1][2*jj]=Ka[2*ij+1][2*jj]+kk[1][2];
Ka[2*ij+1][2*jj+1]=Ka[2*ij+1][2*jj+1]+kk[1][3];
Ka[2*ij+1][2*kj]=Ka[2*ij+1][2*kj]+kk[1][4];
Ka[2*ij+1][2*kj+1]=Ka[2*ij+1][2*kj+1]+kk[1][5];

Ka[2*jj][2*ij]=Ka[2*jj][2*ij]+kk[2][0];
Ka[2*jj][2*ij+1]=Ka[2*jj][2*ij+1]+kk[2][1];
Ka[2*jj][2*jj]=Ka[2*jj][2*jj]+kk[2][2];
Ka[2*jj][2*jj+1]=Ka[2*jj][2*jj+1]+kk[2][3];
Ka[2*jj][2*kj]=Ka[2*jj][2*kj]+kk[2][4];
Ka[2*jj][2*kj+1]=Ka[2*jj][2*kj+1]+kk[2][5];

Ka[2*jj+1][2*ij]=Ka[2*jj+1][2*ij]+kk[3][0];
Ka[2*jj+1][2*ij+1]=Ka[2*jj+1][2*ij+1]+kk[3][1];
Ka[2*jj+1][2*jj]=Ka[2*jj+1][2*jj]+kk[3][2];
Ka[2*jj+1][2*jj+1]=Ka[2*jj+1][2*jj+1]+kk[3][3];
Ka[2*jj+1][2*kj]=Ka[2*jj+1][2*kj]+kk[3][4];
Ka[2*jj+1][2*kj+1]=Ka[2*jj+1][2*kj+1]+kk[3][5];

Ka[2*kj][2*ij]=Ka[2*kj][2*ij]+kk[4][0];
Ka[2*kj][2*ij+1]=Ka[2*kj][2*ij+1]+kk[4][1];
Ka[2*kj][2*jj]=Ka[2*kj][2*jj]+kk[4][2];
Ka[2*kj][2*jj+1]=Ka[2*kj][2*jj+1]+kk[4][3];

```

---

---

```

    Ka[2*kj][2*kj]=Ka[2*kj][2*kj]+kk[4][4];
    Ka[2*kj][2*kj+1]=Ka[2*kj][2*kj+1]+kk[4][5];

    Ka[2*kj+1][2*ij]=Ka[2*kj+1][2*ij]+kk[5][0];
    Ka[2*kj+1][2*ij+1]=Ka[2*kj+1][2*ij+1]+kk[5][1];
    Ka[2*kj+1][2*jj]=Ka[2*kj+1][2*jj]+kk[5][2];
    Ka[2*kj+1][2*jj+1]=Ka[2*kj+1][2*jj+1]+kk[5][3];
    Ka[2*kj+1][2*kj]=Ka[2*kj+1][2*kj]+kk[5][4];
    Ka[2*kj+1][2*kj+1]=Ka[2*kj+1][2*kj+1]+kk[5][5];
  }
}

dhmovek()
{
  int i, j, k, l, m, ii, ji, ki, ij, jj, kj;
  double a, y, x, sq1, sq2, em, E, Nu;
  double bb[8][8], dd[8][8], kk[8][8], kd[8][8];

  for(i=0;i<170;i++)
  {
    for(j=0;j<170;j++)
    {
      Ka[i][j]=0;
    }
  }

  for(i=0;i<meshy2-1;i++)
  {
    for(j=0;j<meshx2-1;j++)
    {
      if(meshx2 < 5)
      {
        E=E1;
        Nu=Nu1;
      }
      else
      {
        E=E2;
        Nu=Nu2;
      }
      ii=i*meshx2+j; ij=i*meshx2+j+1;
      ji=i*meshx2+j+1; kj=(i+1)*meshx2+j;
      ki=(i+1)*meshx2+j; jj=(i+1)*meshx2+j+1;
      sq1= pi*(r2+(double)j)/36;
      sq2= pi*(r2+1+(double)j)/36;
      for(k=0;k<7;k++)
      {
        for(l=0;l<7;l++)
        {
          bb[k][l]=kk[k][l]=kd[k][l]=dd[k][l]=0;
        }
      }
    }
  }
}

```

---

```

    }
    a=2*pi*((double)(j+r2))/36;
    bb[0][0]=bb[2][1]=(a/(2*sq1))*(-1));
    bb[1][1]=bb[2][0]=(1/(2*sq1))*(-1));
    bb[0][2]=bb[2][3]=a/(2*sq1);
    bb[1][5]=bb[2][4]=1/(2*sq1);
    bb[0][4]=bb[2][5]=0;
    bb[1][3]=bb[2][2]=0;

    dd[0][0]=dd[1][1]=E/(1-(Nu*Nu));
    dd[1][0]=dd[0][1]=E*Nu/(1-(Nu*Nu));
    dd[0][2]=dd[1][2]=dd[2][0]=dd[2][1]=0;
    dd[2][2]=E/(2*(1+Nu));

    for(k=0;k<6;k++)
    {
        for(l=0;l<3;l++)
        {
            for(m=0;m<3;m++)
            {
                kd[l][k]=kd[l][k]+dd[l][m]*bb[m][k];
            }
        }
    }

    for(k=0;k<6;k++)
    {
        for(l=0;l<6;l++)
        {
            for(m=0;m<3;m++)
            {
                kk[k][l]=kk[k][l]+2*t*sq1*bb[m][k]*kd[m][l];
            }
        }
    }

    Ka[2*ii][2*ii]=Ka[2*ii][2*ii]+kk[0][0];
    Ka[2*ii][2*ii+1]=Ka[2*ii][2*ii+1]+kk[0][1];
    Ka[2*ii][2*ji]=Ka[2*ii][2*ji]+kk[0][2];
    Ka[2*ii][2*ji+1]=Ka[2*ii][2*ji+1]+kk[0][3];
    Ka[2*ii][2*ki]=Ka[2*ii][2*ki]+kk[0][4];
    Ka[2*ii][2*ki+1]=Ka[2*ii][2*ki+1]+kk[0][5];

    Ka[2*ii+1][2*ii]=Ka[2*ii+1][2*ii]+kk[1][0];
    Ka[2*ii+1][2*ii+1]=Ka[2*ii+1][2*ii+1]+kk[1][1];
    Ka[2*ii+1][2*ji]=Ka[2*ii+1][2*ji]+kk[1][2];
    Ka[2*ii+1][2*ji+1]=Ka[2*ii+1][2*ji+1]+kk[1][3];
    Ka[2*ii+1][2*ki]=Ka[2*ii+1][2*ki]+kk[1][4];
    Ka[2*ii+1][2*ki+1]=Ka[2*ii+1][2*ki+1]+kk[1][5];

    Ka[2*ji][2*ii]=Ka[2*ji][2*ii]+kk[2][0];
    Ka[2*ji][2*ii+1]=Ka[2*ji][2*ii+1]+kk[2][1];
    Ka[2*ji][2*ji]=Ka[2*ji][2*ji]+kk[2][2];
    Ka[2*ji][2*ji+1]=Ka[2*ji][2*ji+1]+kk[2][3];
    Ka[2*ji][2*ki]=Ka[2*ji][2*ki]+kk[2][4];
    Ka[2*ji][2*ki+1]=Ka[2*ji][2*ki+1]+kk[2][5];

```

```

Ka[2*ji+1][2*ii]=Ka[2*ji+1][2*ii]+kk[3][0];
Ka[2*ji+1][2*ii+1]=Ka[2*ji+1][2*ii+1]+kk[3][1];
Ka[2*ji+1][2*ji]=Ka[2*ji+1][2*ji]+kk[3][2];
Ka[2*ji+1][2*ji+1]=Ka[2*ji+1][2*ji+1]+kk[3][3];
Ka[2*ji+1][2*ki]=Ka[2*ji+1][2*ki]+kk[3][4];
Ka[2*ji+1][2*ki+1]=Ka[2*ji+1][2*ki+1]+kk[3][5];

```

```

Ka[2*ki][2*ii]=Ka[2*ki][2*ii]+kk[4][0];
Ka[2*ki][2*ii+1]=Ka[2*ki][2*ii+1]+kk[4][1];
Ka[2*ki][2*ji]=Ka[2*ki][2*ji]+kk[4][2];
Ka[2*ki][2*ji+1]=Ka[2*ki][2*ji+1]+kk[4][3];
Ka[2*ki][2*ki]=Ka[2*ki][2*ki]+kk[4][4];
Ka[2*ki][2*ki+1]=Ka[2*ki][2*ki+1]+kk[4][5];

```

```

Ka[2*ki+1][2*ii]=Ka[2*ki+1][2*ii]+kk[5][0];
Ka[2*ki+1][2*ii+1]=Ka[2*ki+1][2*ii+1]+kk[5][1];
Ka[2*ki+1][2*ji]=Ka[2*ki+1][2*ji]+kk[5][2];
Ka[2*ki+1][2*ji+1]=Ka[2*ki+1][2*ji+1]+kk[5][3];
Ka[2*ki+1][2*ki]=Ka[2*ki+1][2*ki]+kk[5][4];
Ka[2*ki+1][2*ki+1]=Ka[2*ki+1][2*ki+1]+kk[5][5];

```

```

for(k=0;k<7;k++)
{
    for(l=0;l<7;l++)
    {
        bb[k][l]=kk[k][l]=kd[k][l]=0;
    }
}

```

```

a=2*pi*((double)(r2+j+1))/36;
bb[0][0]=bb[2][1]=0;
bb[1][1]=bb[2][0]=(1/(2*sq2)*(-1));
bb[0][2]=bb[2][3]=a/(2*sq2);
bb[1][5]=bb[2][4]=0;
bb[0][4]=bb[2][5]=(a/(2*sq2)*(-1));
bb[1][3]=bb[2][2]=1/(2*sq2);

```

```

for(k=0;k<6;k++)
{
    for(l=0;l<3;l++)
    {
        for(m=0;m<3;m++)
        {
            kd[l][k]=kd[l][k]+dd[l][m]*bb[m][k];
        }
    }
}

```

```

for(k=0;k<6;k++)
{
    for(l=0;l<6;l++)
    {
        for(m=0;m<3;m++)
        {
            kk[k][l]=kk[k][l]+2*i*sq2*bb[m][k]*kd[m][l];
        }
    }
}

```

---

```

        }
    }
    Ka[2*ij][2*ij]=Ka[2*ij][2*ij]+kk[0][0];
    Ka[2*ij][2*ij+1]=Ka[2*ij][2*ij+1]+kk[0][1];
    Ka[2*ij][2*jj]=Ka[2*ij][2*jj]+kk[0][2];
    Ka[2*ij][2*jj+1]=Ka[2*ij][2*jj+1]+kk[0][3];
    Ka[2*ij][2*kj]=Ka[2*ij][2*kj]+kk[0][4];
    Ka[2*ij][2*kj+1]=Ka[2*ij][2*kj+1]+kk[0][5];

    Ka[2*ij+1][2*ij]=Ka[2*ij+1][2*ij]+kk[1][0];
    Ka[2*ij+1][2*ij+1]=Ka[2*ij+1][2*ij+1]+kk[1][1];
    Ka[2*ij+1][2*jj]=Ka[2*ij+1][2*jj]+kk[1][2];
    Ka[2*ij+1][2*jj+1]=Ka[2*ij+1][2*jj+1]+kk[1][3];
    Ka[2*ij+1][2*kj]=Ka[2*ij+1][2*kj]+kk[1][4];
    Ka[2*ij+1][2*kj+1]=Ka[2*ij+1][2*kj+1]+kk[1][5];

    Ka[2*jj][2*ij]=Ka[2*jj][2*ij]+kk[2][0];
    Ka[2*jj][2*ij+1]=Ka[2*jj][2*ij+1]+kk[2][1];
    Ka[2*jj][2*jj]=Ka[2*jj][2*jj]+kk[2][2];
    Ka[2*jj][2*jj+1]=Ka[2*jj][2*jj+1]+kk[2][3];
    Ka[2*jj][2*kj]=Ka[2*jj][2*kj]+kk[2][4];
    Ka[2*jj][2*kj+1]=Ka[2*jj][2*kj+1]+kk[2][5];

    Ka[2*jj+1][2*ij]=Ka[2*jj+1][2*ij]+kk[3][0];
    Ka[2*jj+1][2*ij+1]=Ka[2*jj+1][2*ij+1]+kk[3][1];
    Ka[2*jj+1][2*jj]=Ka[2*jj+1][2*jj]+kk[3][2];
    Ka[2*jj+1][2*jj+1]=Ka[2*jj+1][2*jj+1]+kk[3][3];
    Ka[2*jj+1][2*kj]=Ka[2*jj+1][2*kj]+kk[3][4];
    Ka[2*jj+1][2*kj+1]=Ka[2*jj+1][2*kj+1]+kk[3][5];

    Ka[2*kj][2*ij]=Ka[2*kj][2*ij]+kk[4][0];
    Ka[2*kj][2*ij+1]=Ka[2*kj][2*ij+1]+kk[4][1];
    Ka[2*kj][2*jj]=Ka[2*kj][2*jj]+kk[4][2];
    Ka[2*kj][2*jj+1]=Ka[2*kj][2*jj+1]+kk[4][3];
    Ka[2*kj][2*kj]=Ka[2*kj][2*kj]+kk[4][4];
    Ka[2*kj][2*kj+1]=Ka[2*kj][2*kj+1]+kk[4][5];

    Ka[2*kj+1][2*ij]=Ka[2*kj+1][2*ij]+kk[5][0];
    Ka[2*kj+1][2*ij+1]=Ka[2*kj+1][2*ij+1]+kk[5][1];
    Ka[2*kj+1][2*jj]=Ka[2*kj+1][2*jj]+kk[5][2];
    Ka[2*kj+1][2*jj+1]=Ka[2*kj+1][2*jj+1]+kk[5][3];
    Ka[2*kj+1][2*kj]=Ka[2*kj+1][2*kj]+kk[5][4];
    Ka[2*kj+1][2*kj+1]=Ka[2*kj+1][2*kj+1]+kk[5][5];

    }
}

/*
intgfor(double e,int i)
{
    char file[64];
    int j, k;
    FILE *fp;

```

---

---

```
j=(int)(e*ficnt+i*ficnt);
itoa(j,file,10);
if(NULL == (fp = fopen(file, "w")))
{
    printf("¥7 ¥n Cannot File Open !",file);
    exit(1);
}
for(k=1;k<9;k++)
{
    fprintf(fp, " %lf",pr[k]);
}
fclose(fp);
}
*/
```

---

## 1 - 1 0 等級による軸の釣合良さの計算

```
#include <stdio.h>
#include <math.h>

#define h1      8
#define h2      8
#define pi      3.1415926
#define m      2.4876
#define b      16
#define l      600
#define s      300
#define c      300
#define a      292

main()
{
    int i, j, rpm;
    double g, e, up, upm, k, r, le;
    double Gin(), Upcal(), Upcal1(), Upcal2();

    le=l;
    k=(le-s)/le;
    r=h1/h2;
    g=Gin();

    printf("\nk=%lf\n",k);

    for(i=0;i<61;i++)
    {
        rpm=1000+(i*100);
        up=Upcal(rpm, g);

        upm=Upcal1(up);

        printf("\n回転数は  %d r p m,  U per =%lf",rpm,up);
        if((upm*(1+r)) < up)
        {
            printf("\n残留不釣合いは U per1condition = %f",upm);
        }
        else
        {
            printf("\n条件を満たしていない U per1condition = %f",upm);
        }

        upm=Upcal2(up, k, r);

        if((upm*(1+r)) < up)
        {
            printf("\n残留不釣合いは U per1free = %f",upm);
        }
        else
        {
            printf("\n条件を満たしていない U per1free = %f",upm);
        }
    }
}
```



```

    }
}

double Gin()
{
double g;

printf("釣り合良さの等級 G を入力");
scanf ("%lf",&g);
return g;
}

double Upcal(int rpm,double g)
{
double up, up1, up3;
double e;

e=g*60*1000/(2*pi*rpm);
up=m*e;
return (up);
}

double Upcal1(double up)
{
double up1, up3, upm;
up1=up*3*1/(8*b);
up3=up*1/(4*c);
if(up1 < up3)
{
upm=up1;
}
else
{
upm=up3;
}

return (upm);
}

double Upcal2(double up,double k,double r)
{
double upm, upm1;

upm=(double)(fabs((float)((up*k*1)/((1-a)+r*(1-a-b)))));
upm1=(double)(fabs((float)((up*k*1)/((1-a)-r*(1-a-b)))));
if(upm1 < upm)
{
upm=upm1;
}
upm1=(double)(fabs((float)((up*(1-k)*1)/(a+r*(a+b)))));
if(upm1 < upm)
{
upm=upm1;
}
upm1=(double)(fabs( (float) ( (up*(1-k)*1)/(a-r*(a+b)) ) ));

```

---

```
if(upm1 < upm)
{
    upm=upm1;
}
```

```
return (upm);
}
```

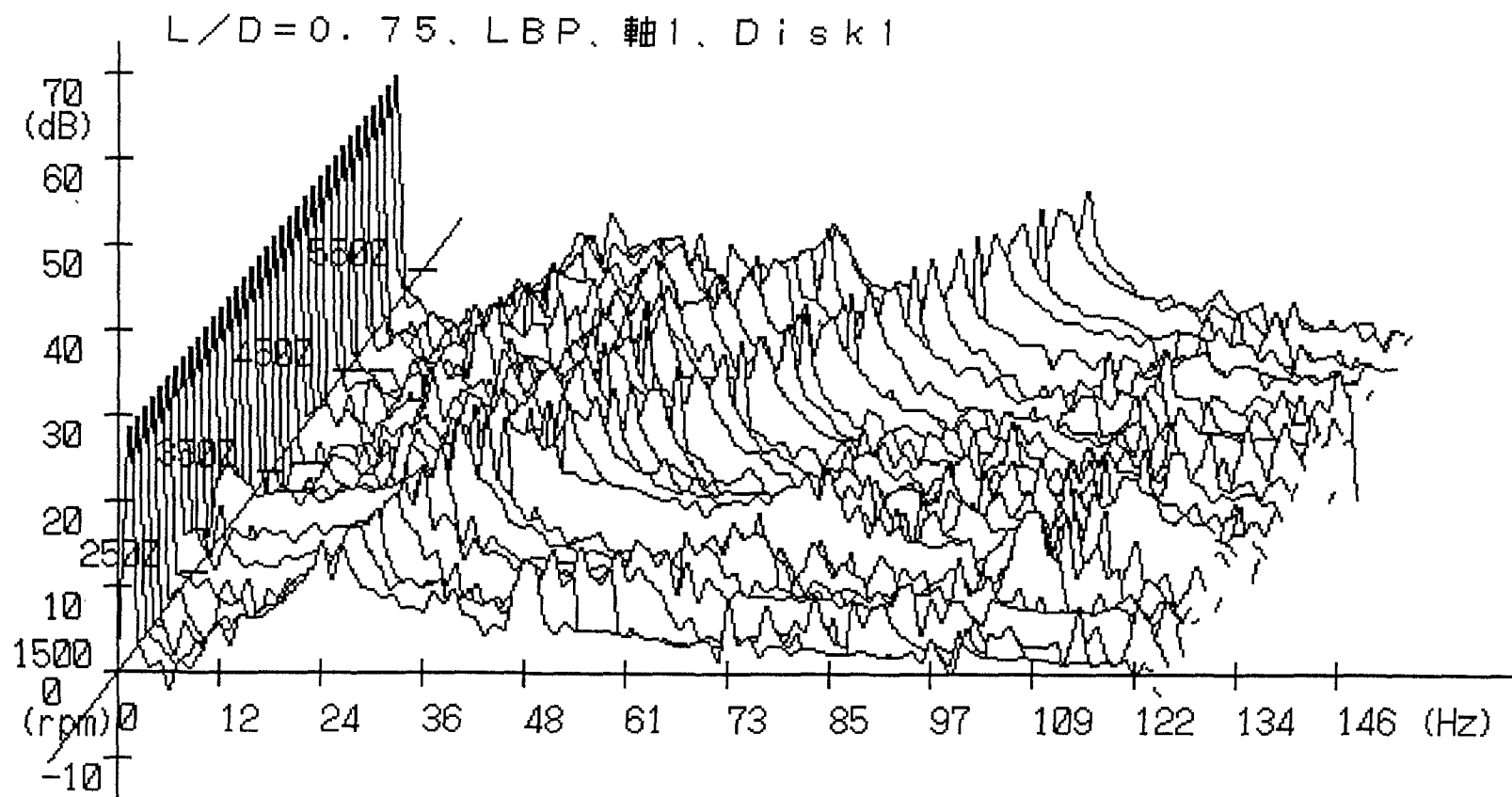
---

## 付録2 非接触変位計の校正結果

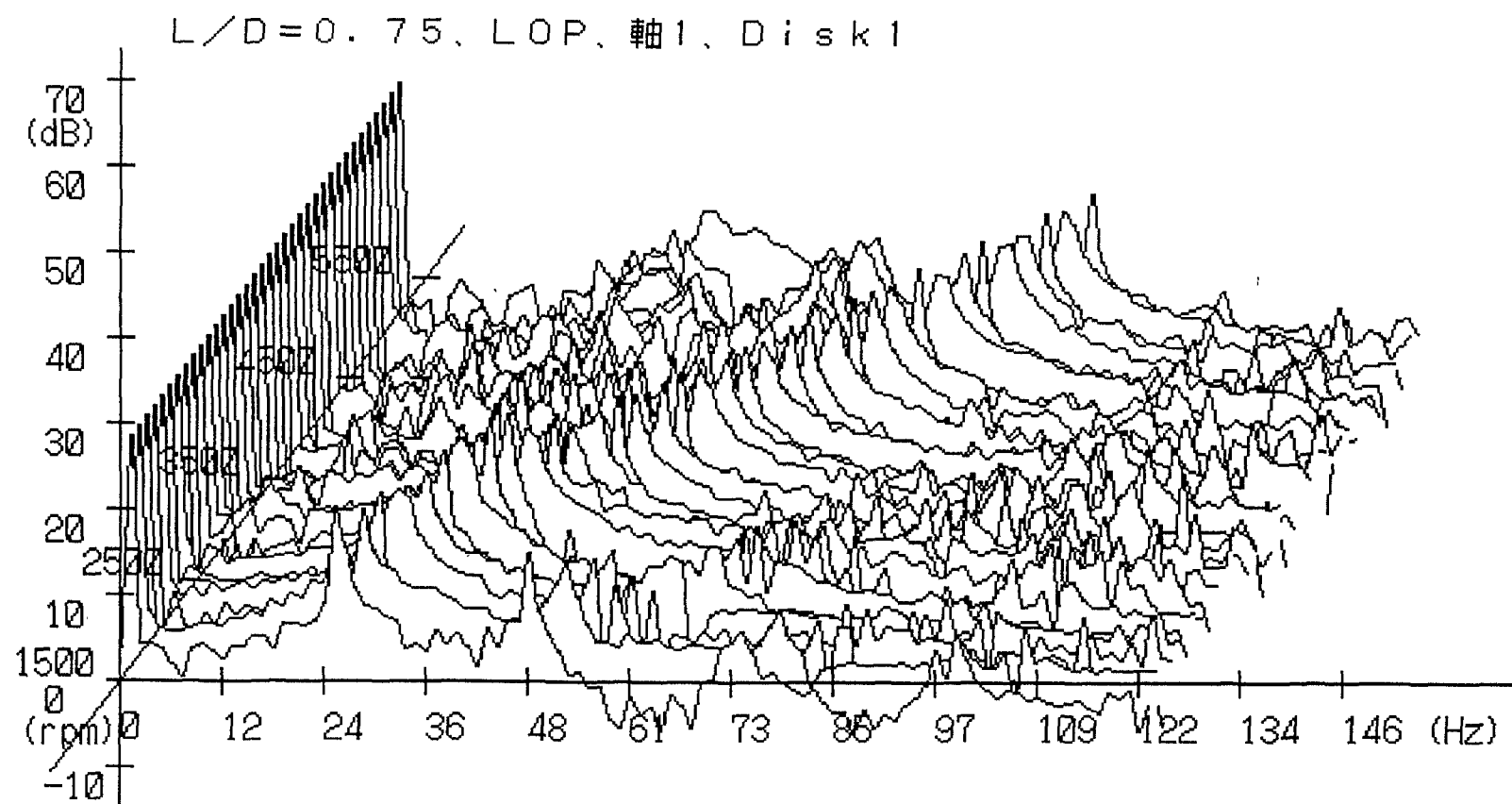
チャンネル 距離 (μm)	7	
	1 6 進数	1 0 進数
50	4dd	1245
100	525	1317
150	566	1382
200	5a2	1442
250	5e0	1504
300	622	1570
350	65e	1630
400	69f	1695
450	6d9	1753
500	719	1817
550	758	1880
600	79c	1948
650	7e0	2016
700	824	2084
750	869	2153
800	8a8	2216
850	8f4	2292
900	93d	2365
950	980	2432
1000	9ca	2506
1050	a1c	2588
1100	a62	2658
1150	aaf	2735
1200	b01	2817
1250	b4e	2894
1300	ba2	2978
1350	bf6	3062
1400	c48	3144
1450	c9f	3231
1500	cf3	3315
1550	d4b	3403
1600	da3	3491
1650	dfc	3580
1700	e58	3672
1750	eb8	3768
1800	f16	3862
1850	f79	3961
1900	fdd	4061
1950	fff	4095

チャンネル 距離 (μm)	6	
	1 6 進数	1 0 進数
2500	fdf	4063
2450	fc4	4036
2400	fa8	4008
2350	f8f	3983
2300	f5c	3932
2250	f41	3905
2200	f24	3876
2150	f01	3841
2100	edc	3804
2050	ebf	3775
2000	e9c	3740
1950	e7a	3706
1900	e54	3668
1850	e2b	3627
1800	e10	3600
1750	de7	3559
1700	dcc	3532
1650	d9c	3484
1550	d56	3414
1500	d2e	3374
1450	d08	3336
1400	ce3	3299
1350	cc8	3272
1300	c95	3221
1250	c6f	3183
1200	c4b	3147
1150	c1d	3101
1100	bff	3071
1050	bef	3023
1000	ba9	2985
950	b73	2931
900	b5b	2907
850	b27	2855
800	af8	2808
750	acd	2765
700	aac	2732
650	a70	2672
600	a43	2627
550	a15	2581
500	9ee	2542
450	9bb	2491
400	98d	2445
350	95f	2399
300	932	2354
250	909	2313
200	8d7	2263
150	8ab	2219
100	87e	2174
50	850	2128

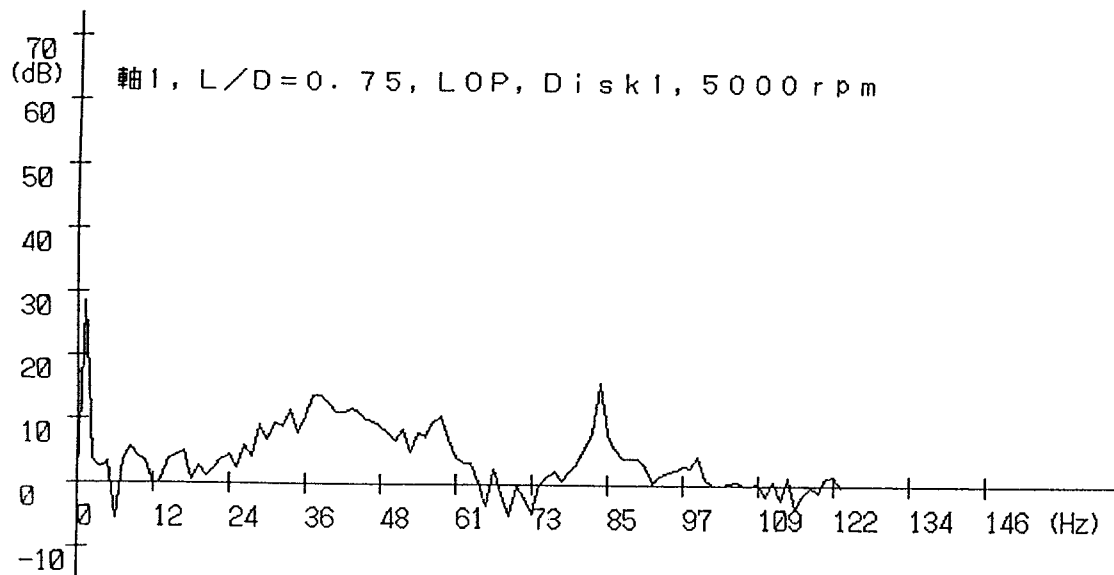
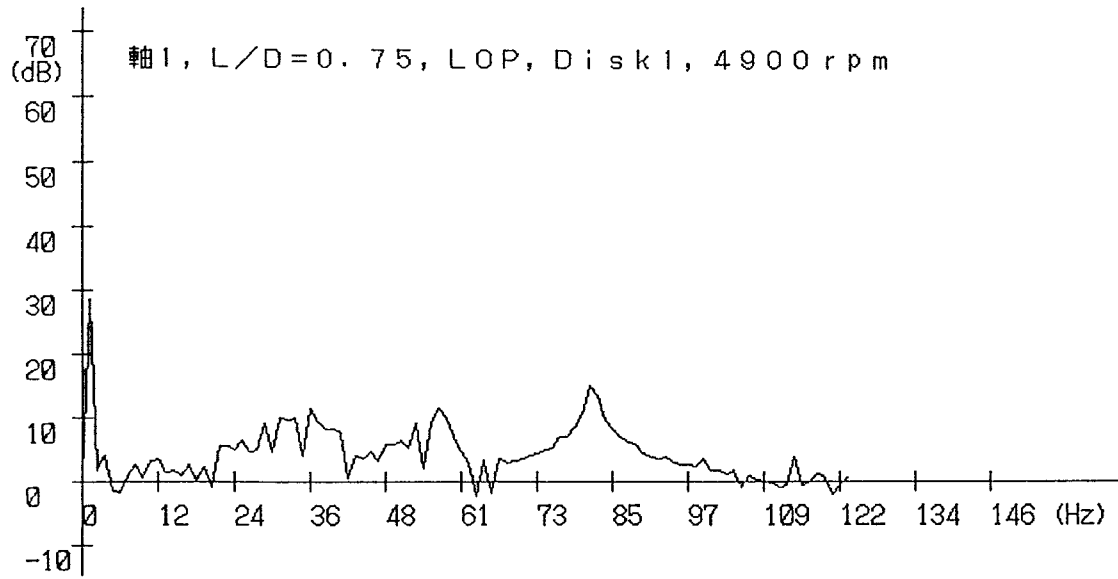
### 付録3 軸の振動解析より求めた安定限界速度



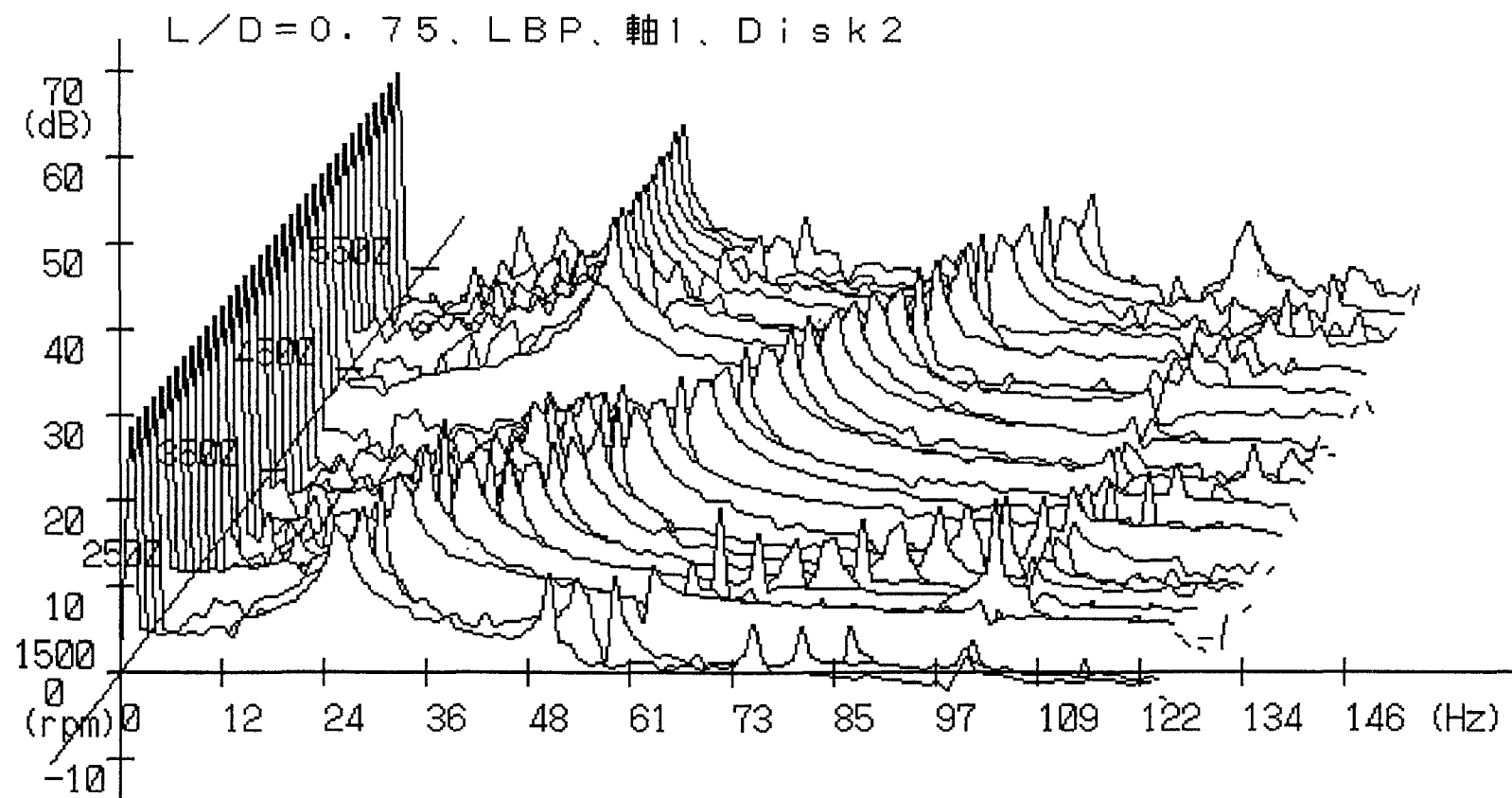
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



回転速度、周波数をパラメータとしたときの振動振幅の解析結果

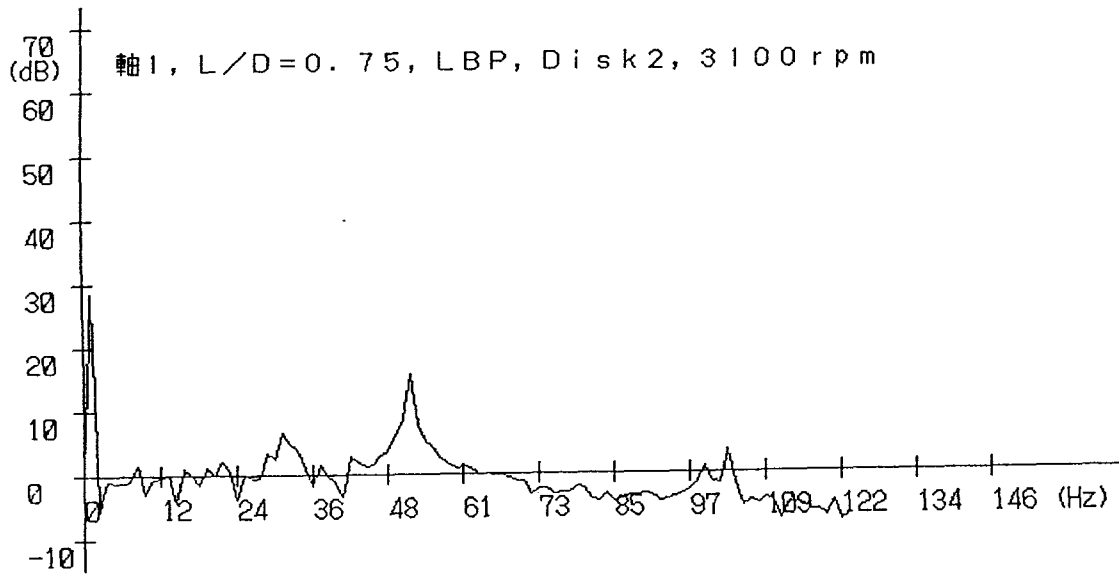
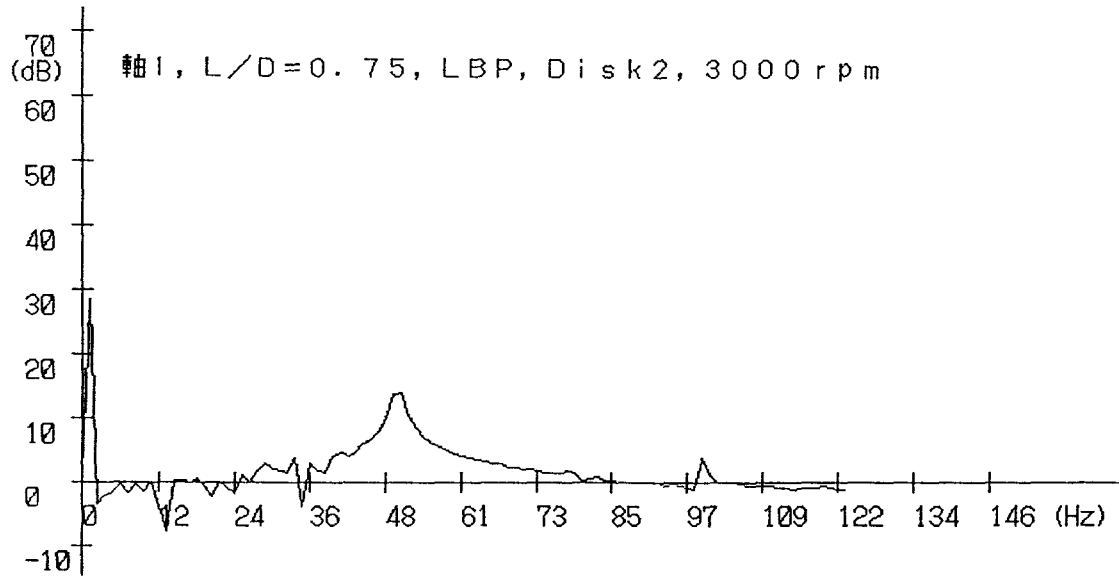


安定限界速度近辺の軸振動振幅

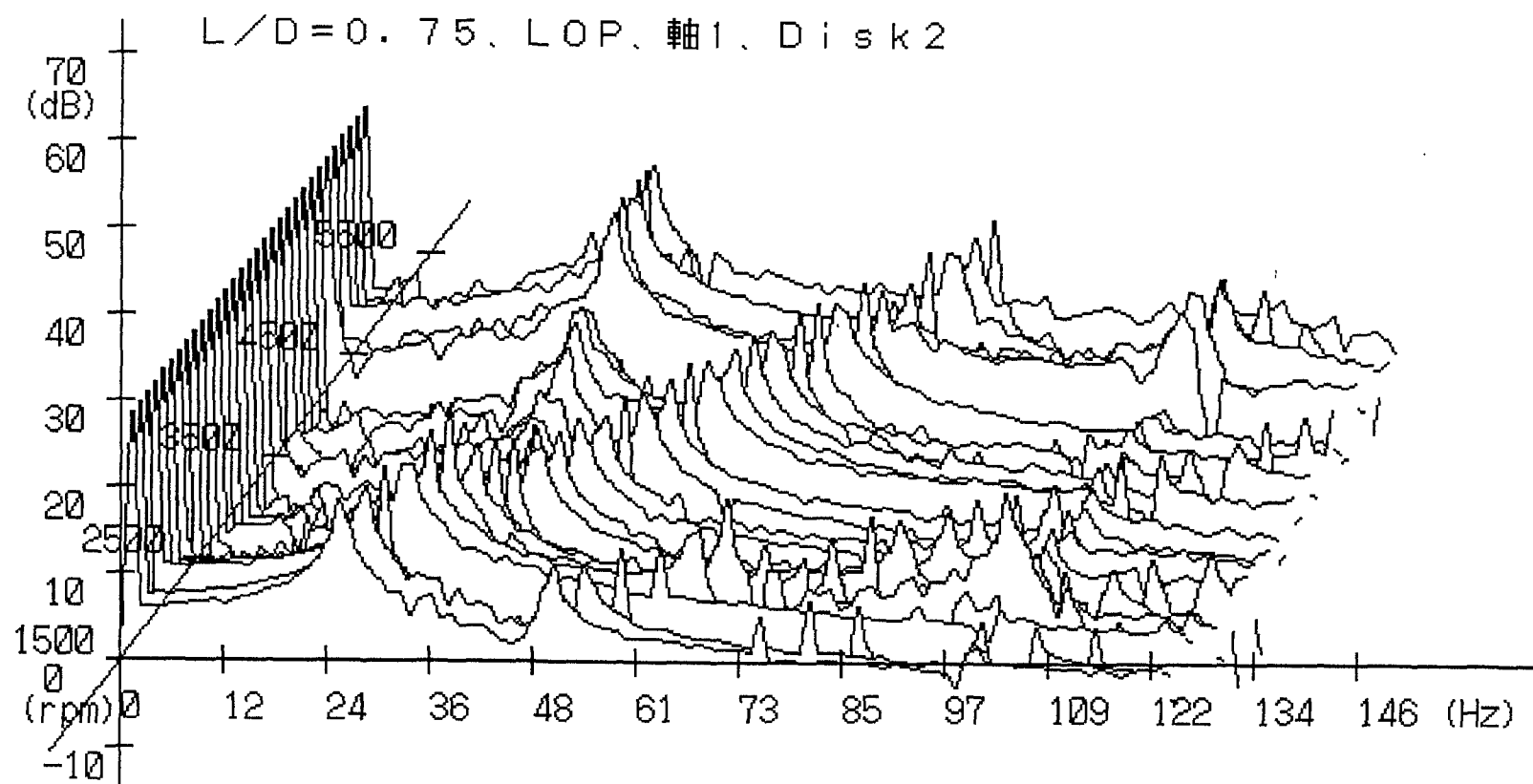


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

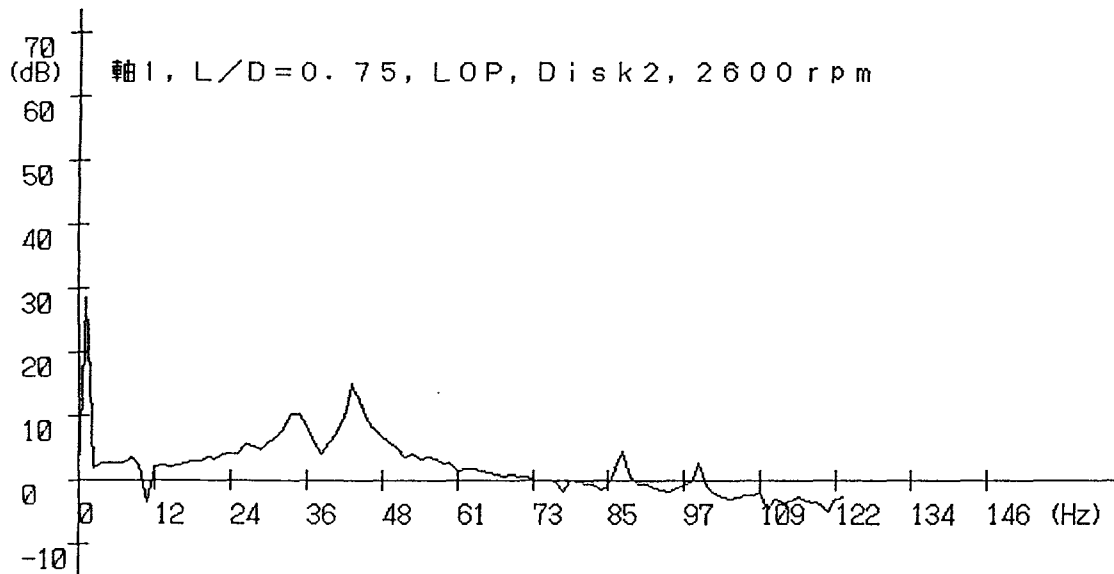
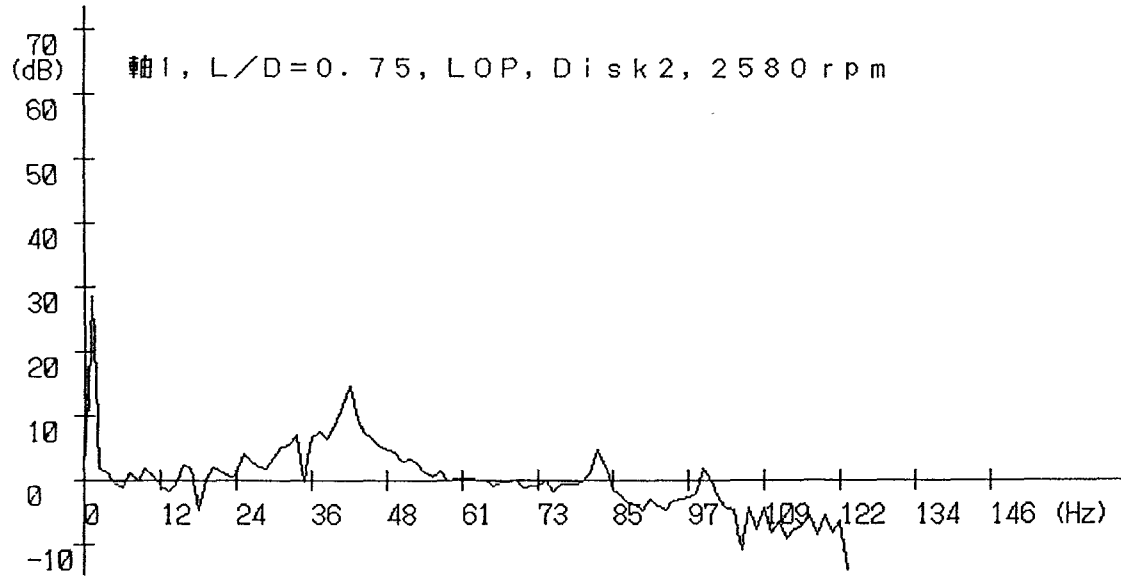




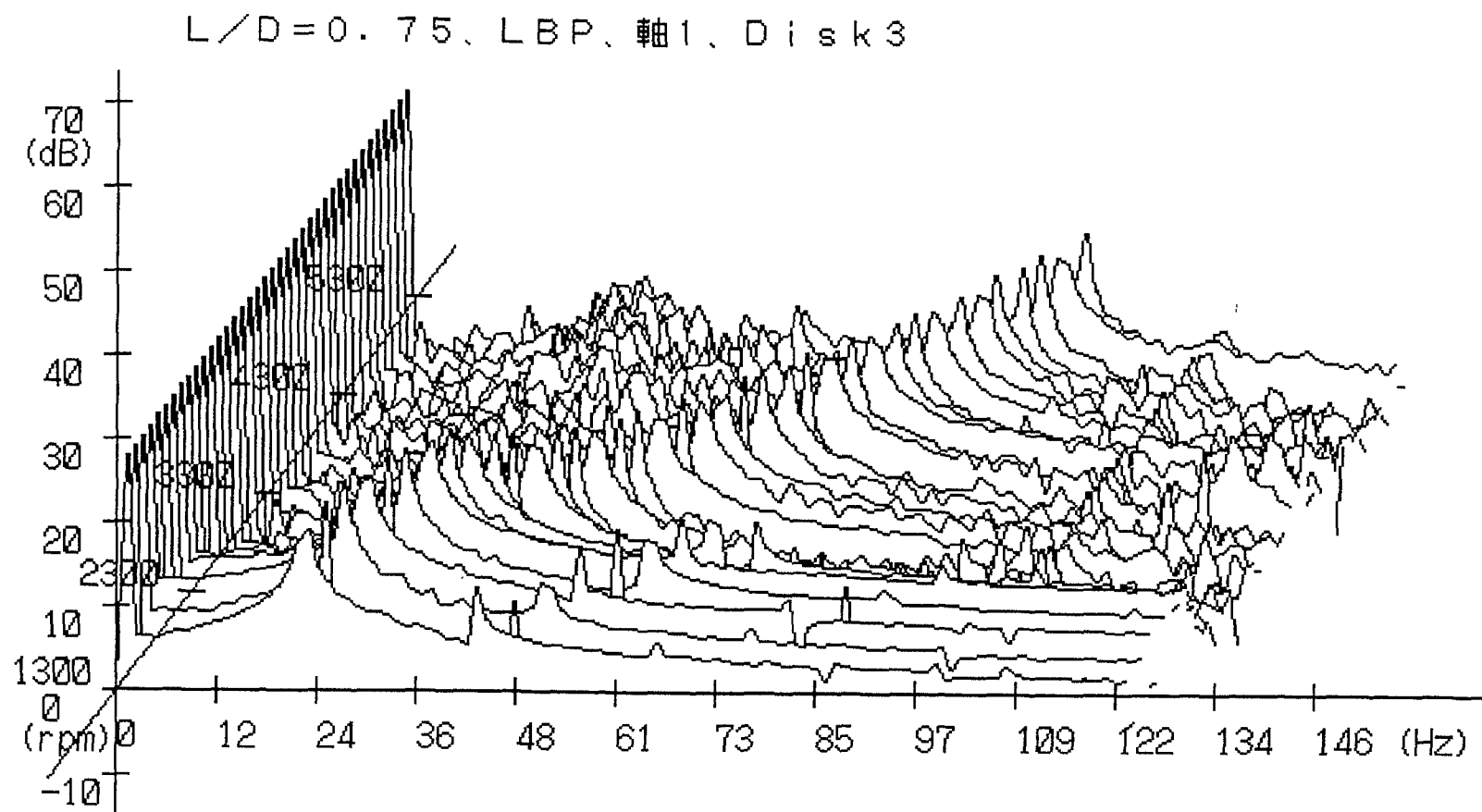
安定限界速度近辺の軸振動振幅



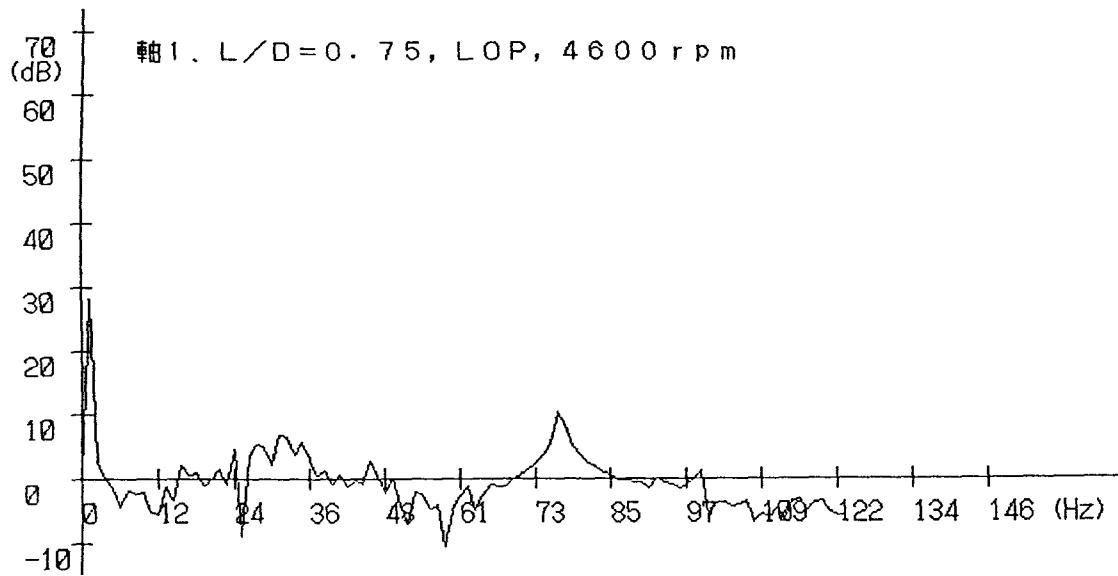
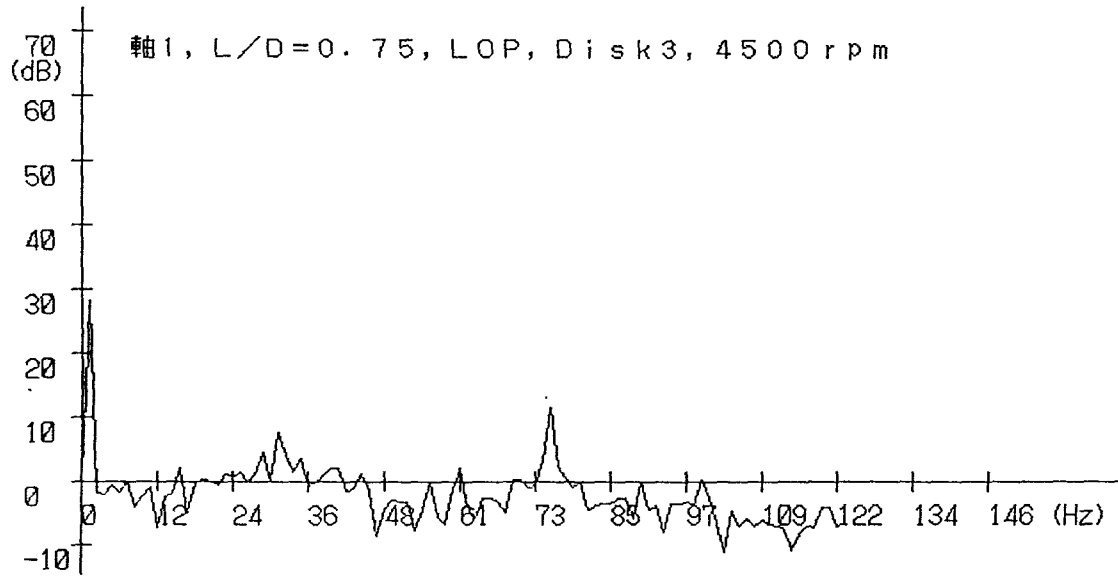
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



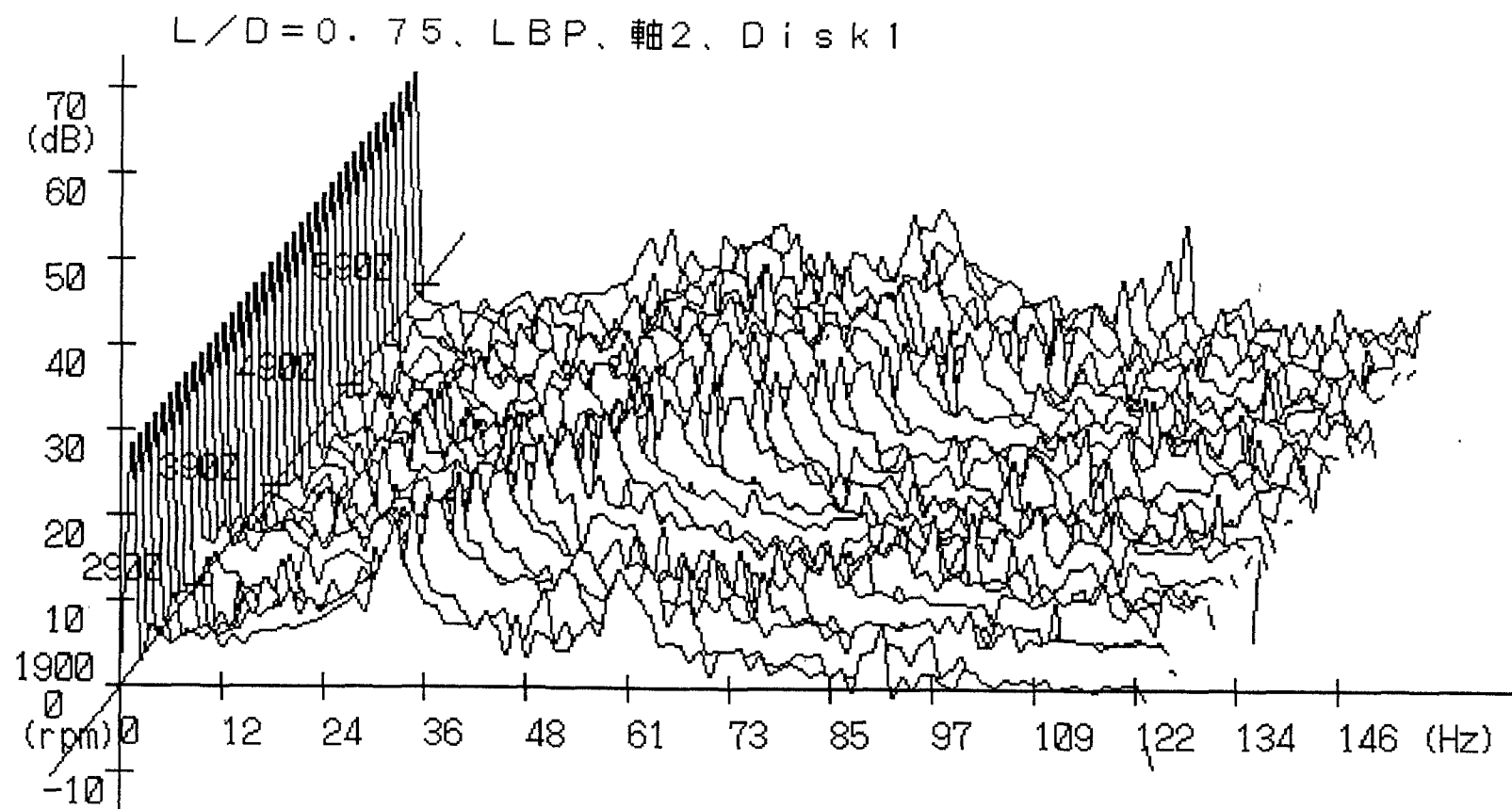
安定限界速度近辺の軸振動振幅



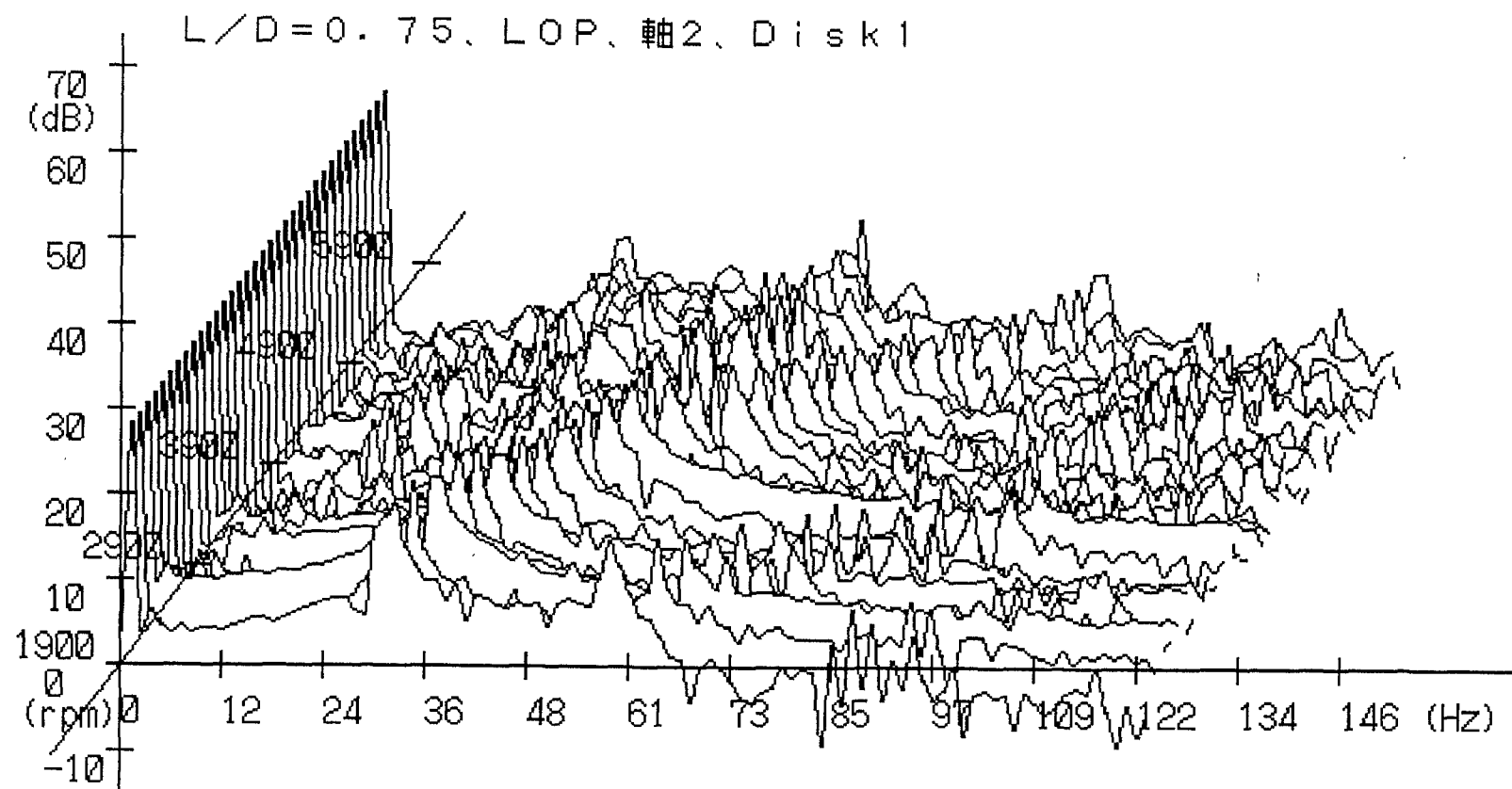
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



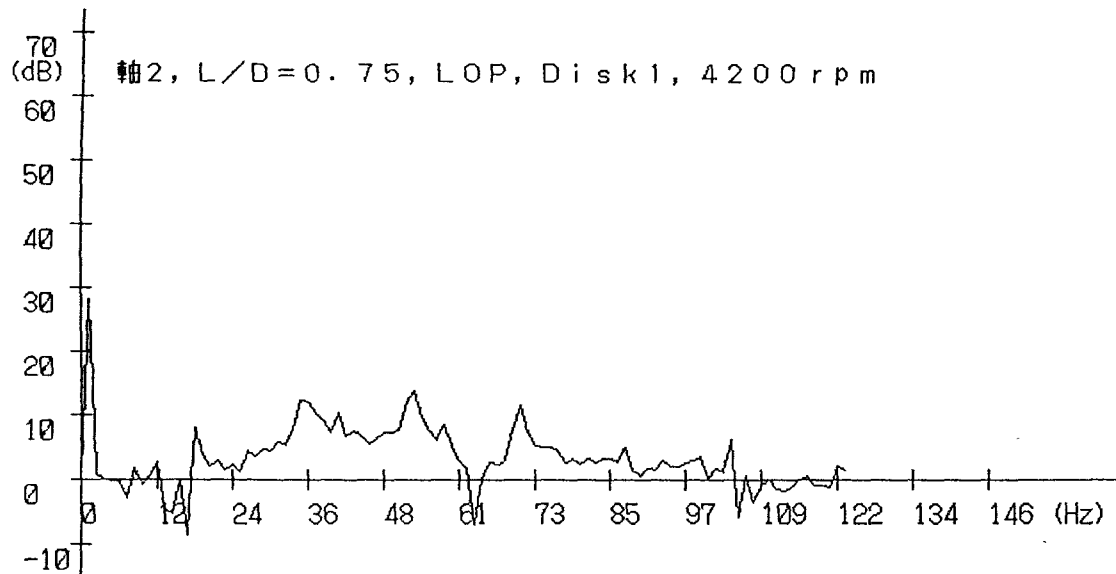
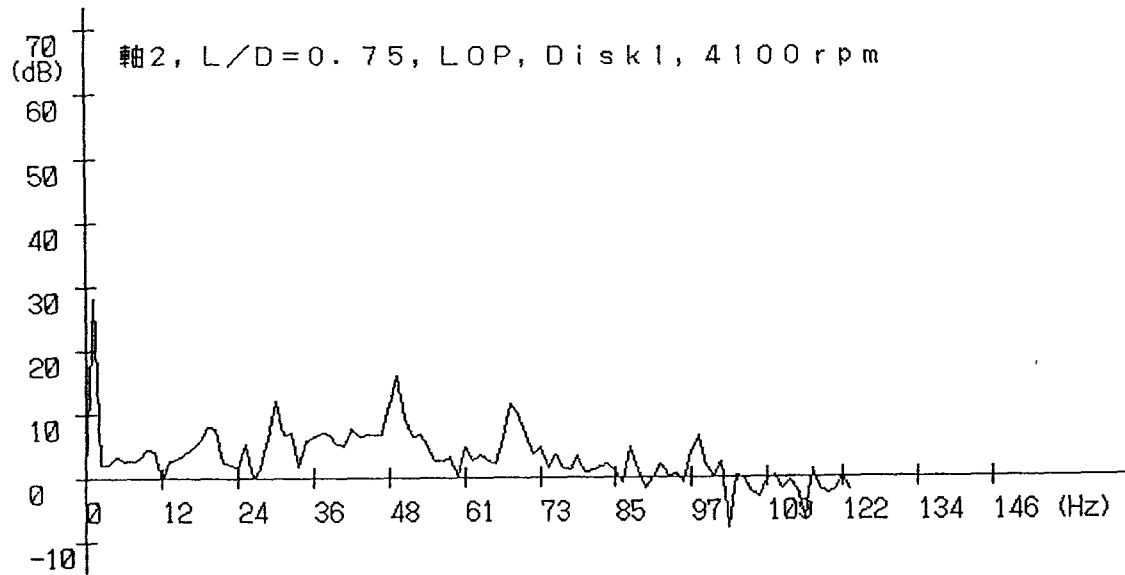
安定限界速度近辺の軸振動振幅



回転速度、周波数をパラメータとしたときの振動振幅の解析結果

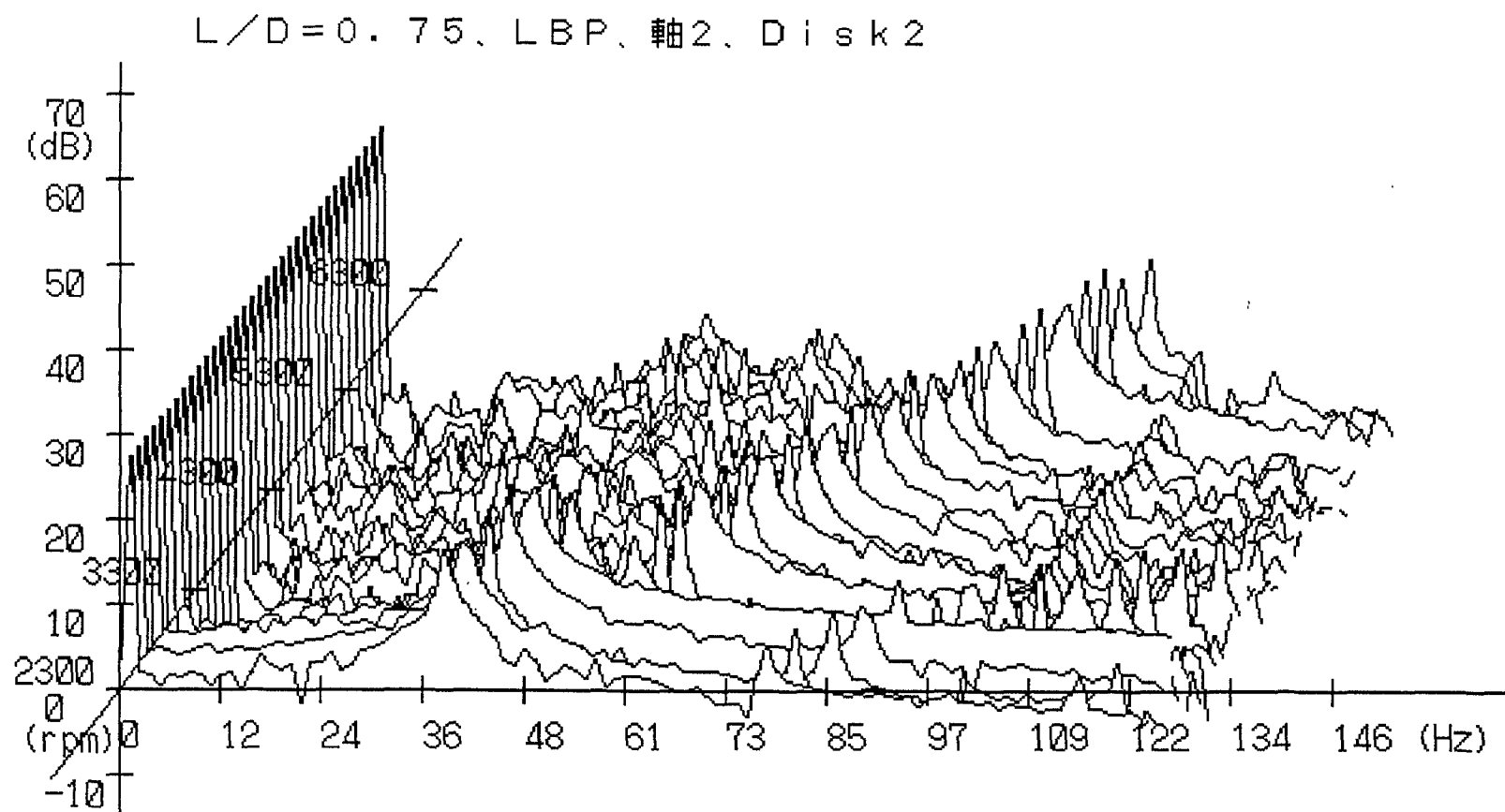


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

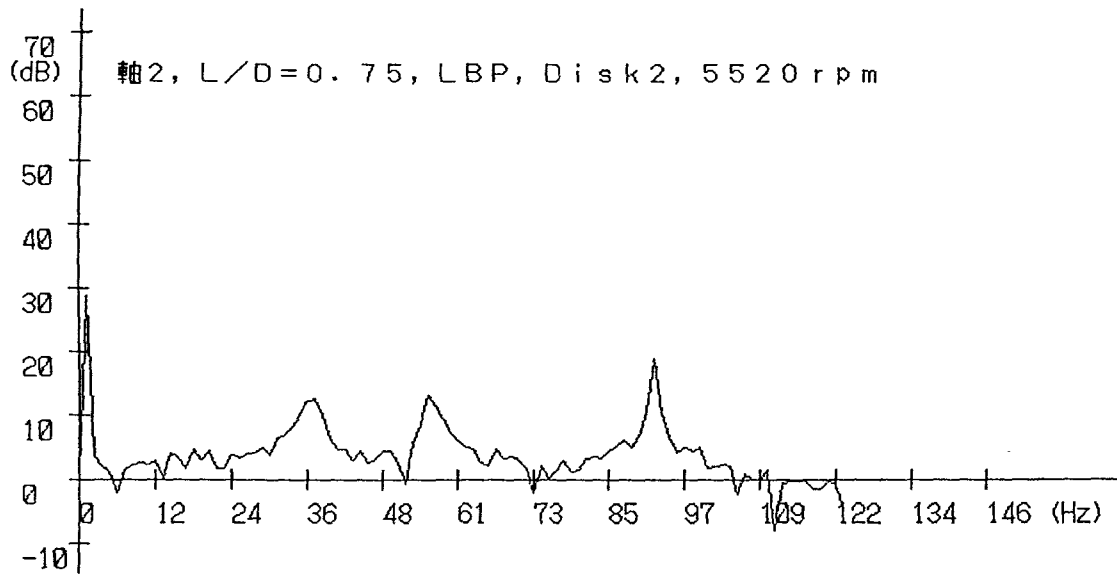
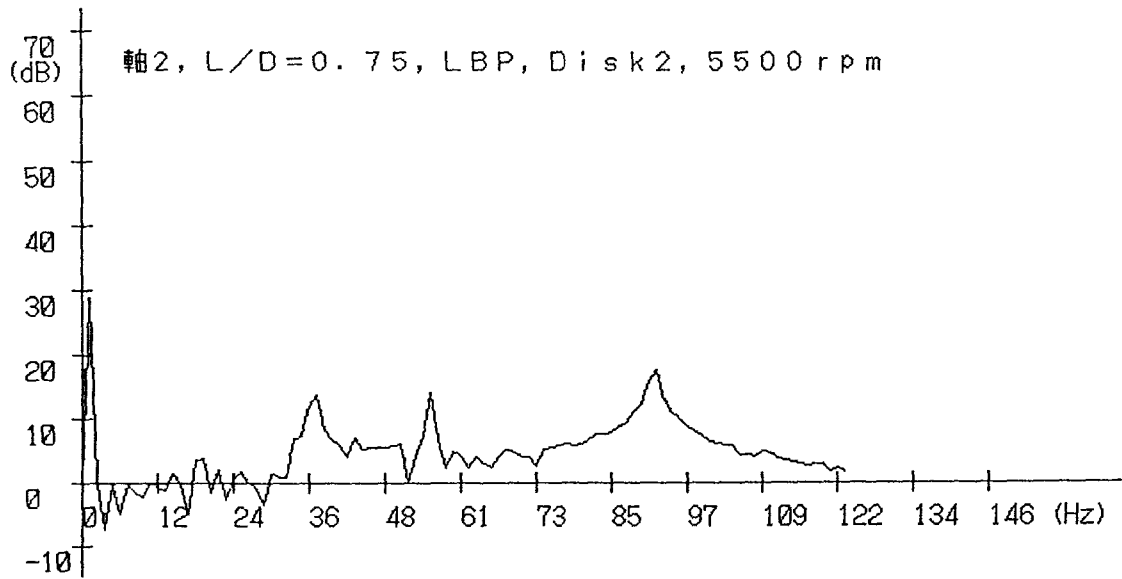


安定限界速度近辺の軸振動振幅



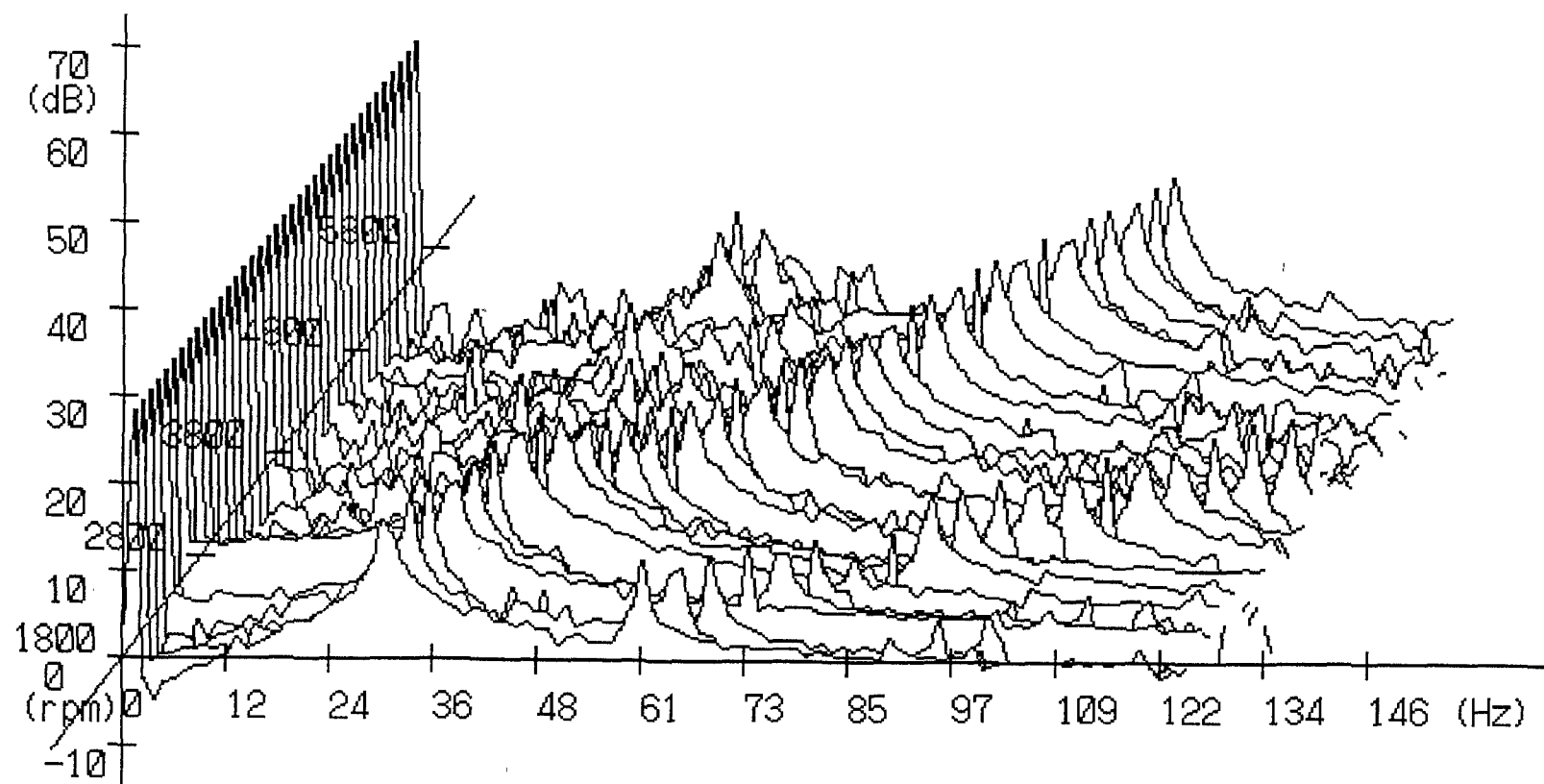


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

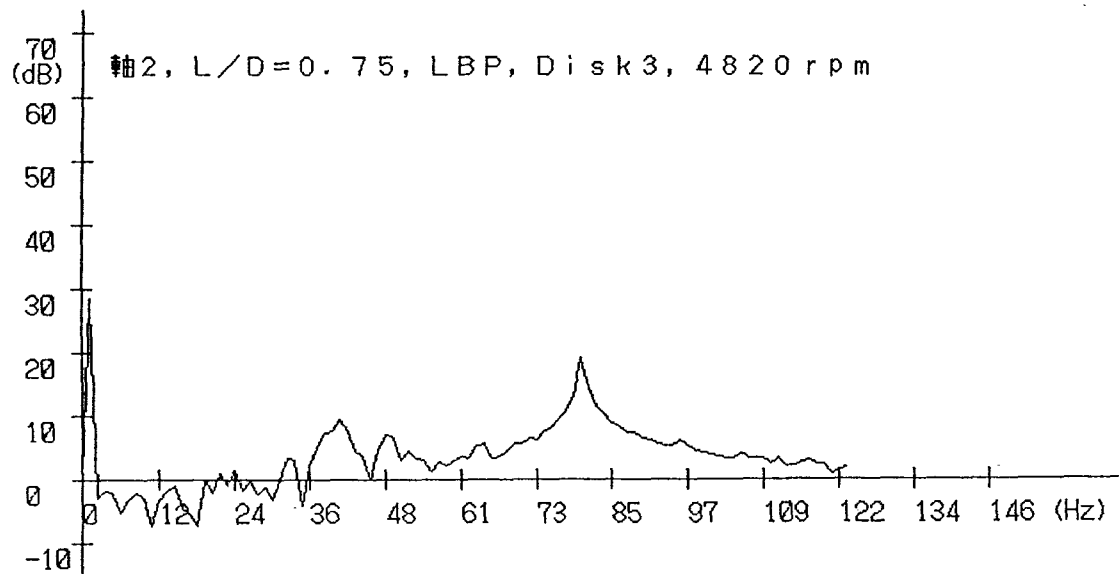
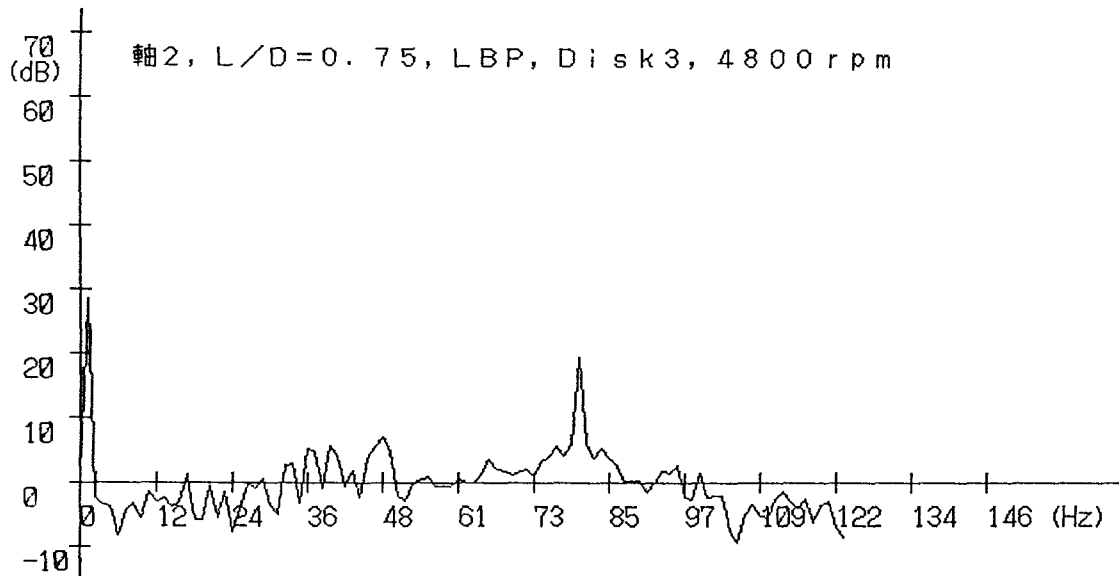


安定限界速度近辺の軸振動振幅

L/D=0.75、LBP、軸2、Disk3

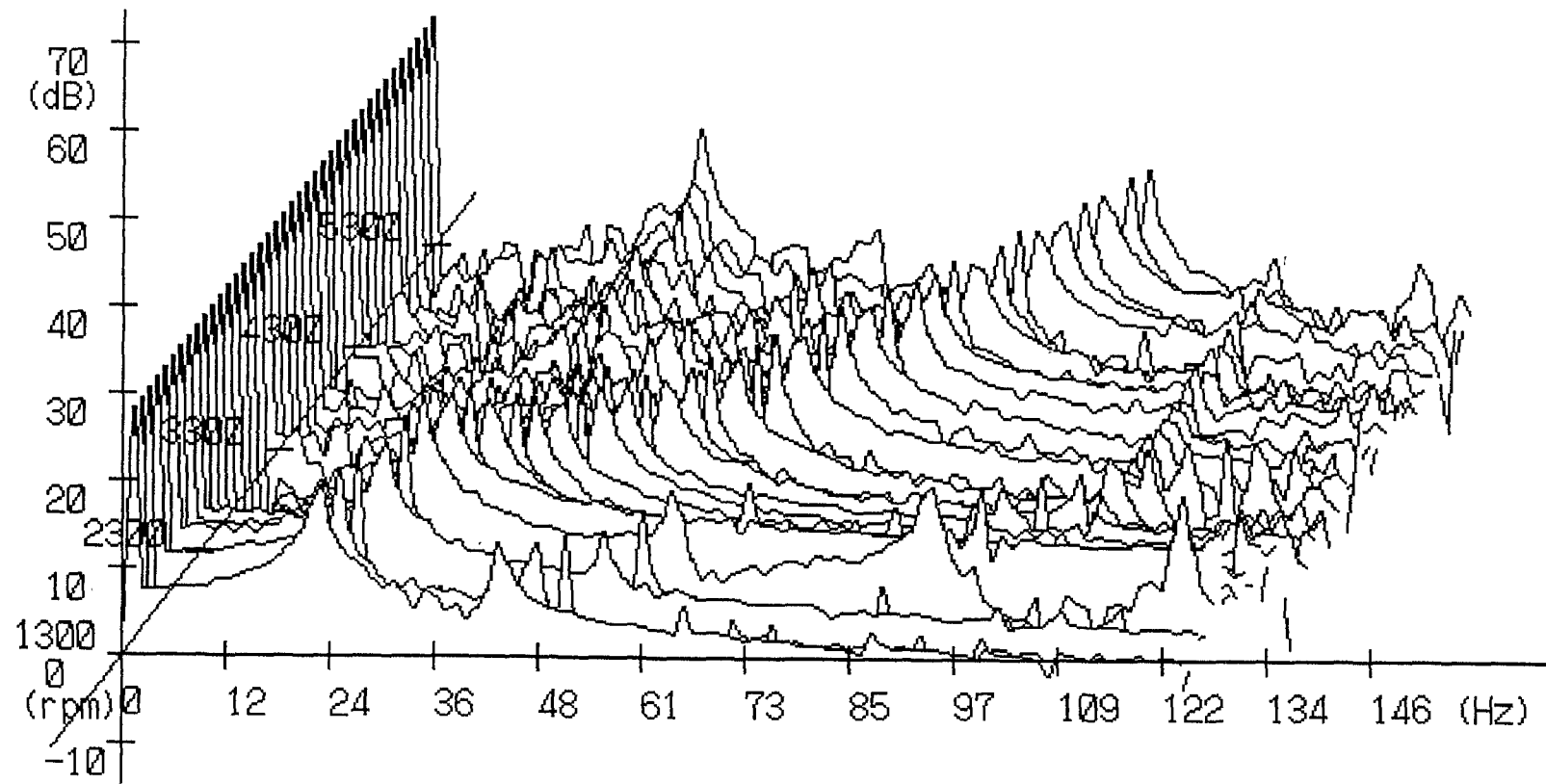


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

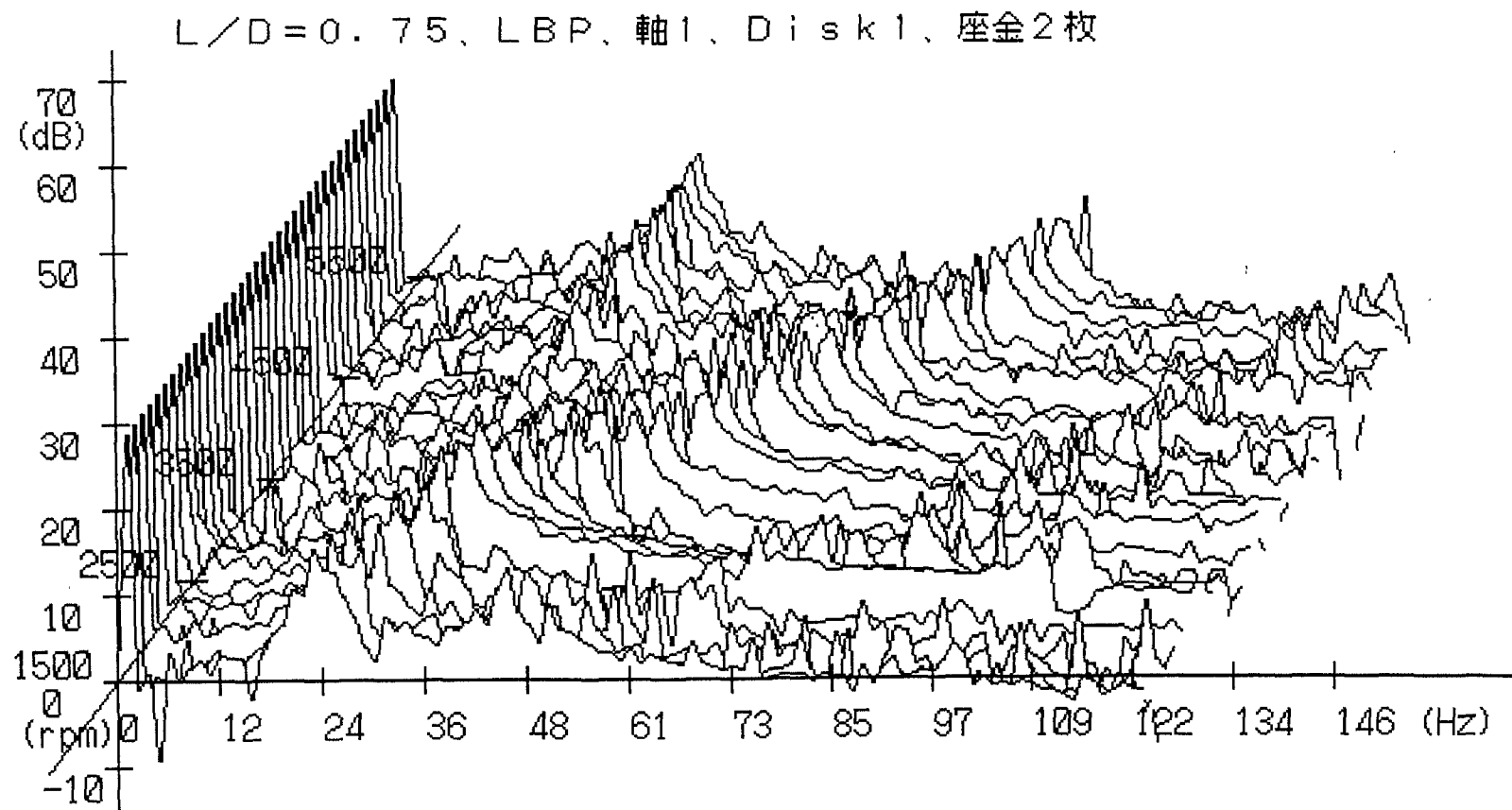


安定限界速度近辺の軸振動振幅

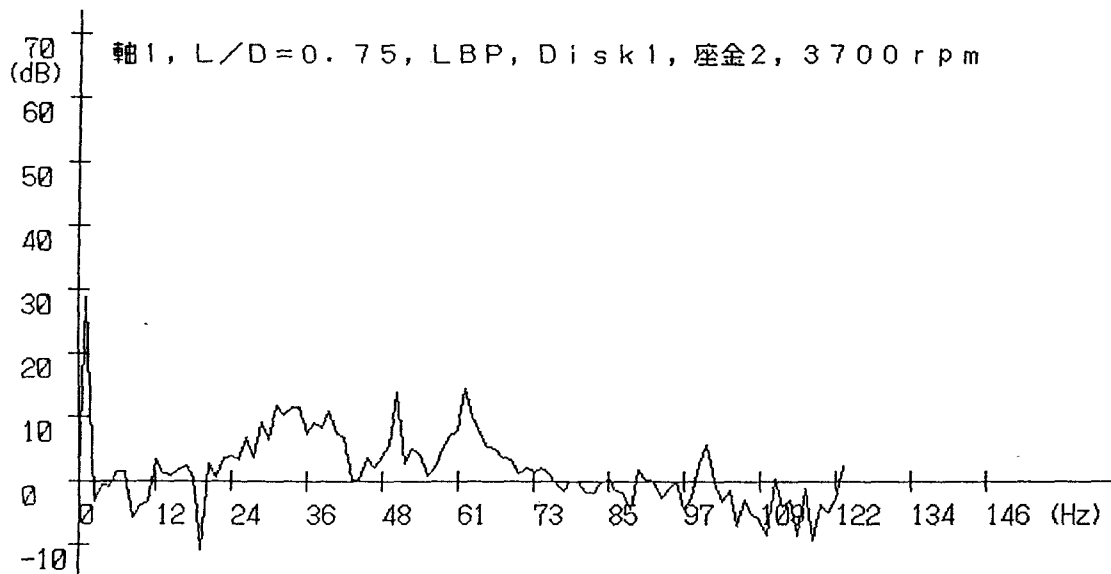
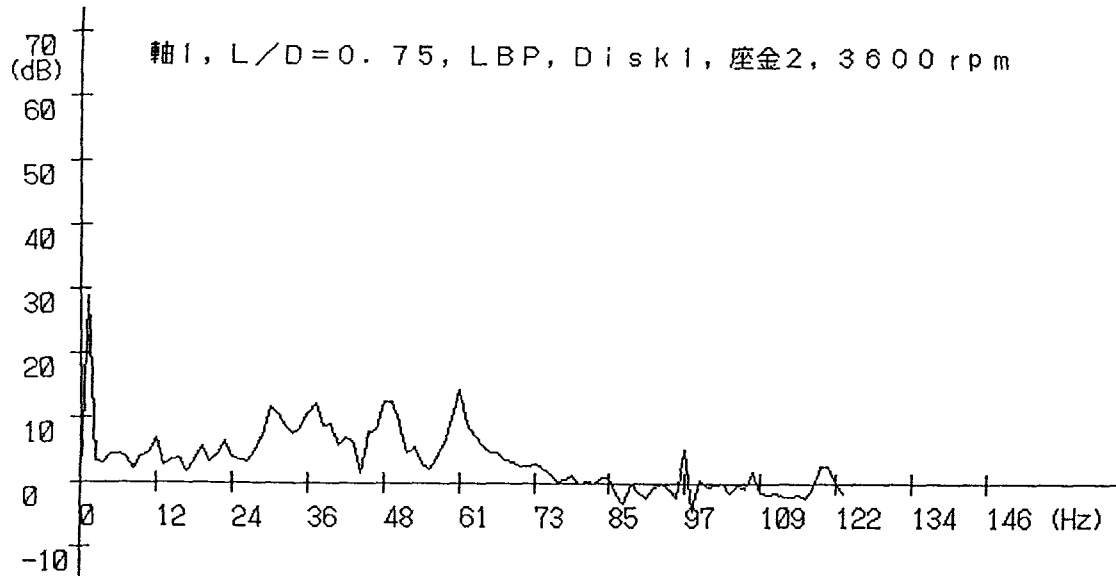
L/D=0.75、LOP、軸1、Disk3



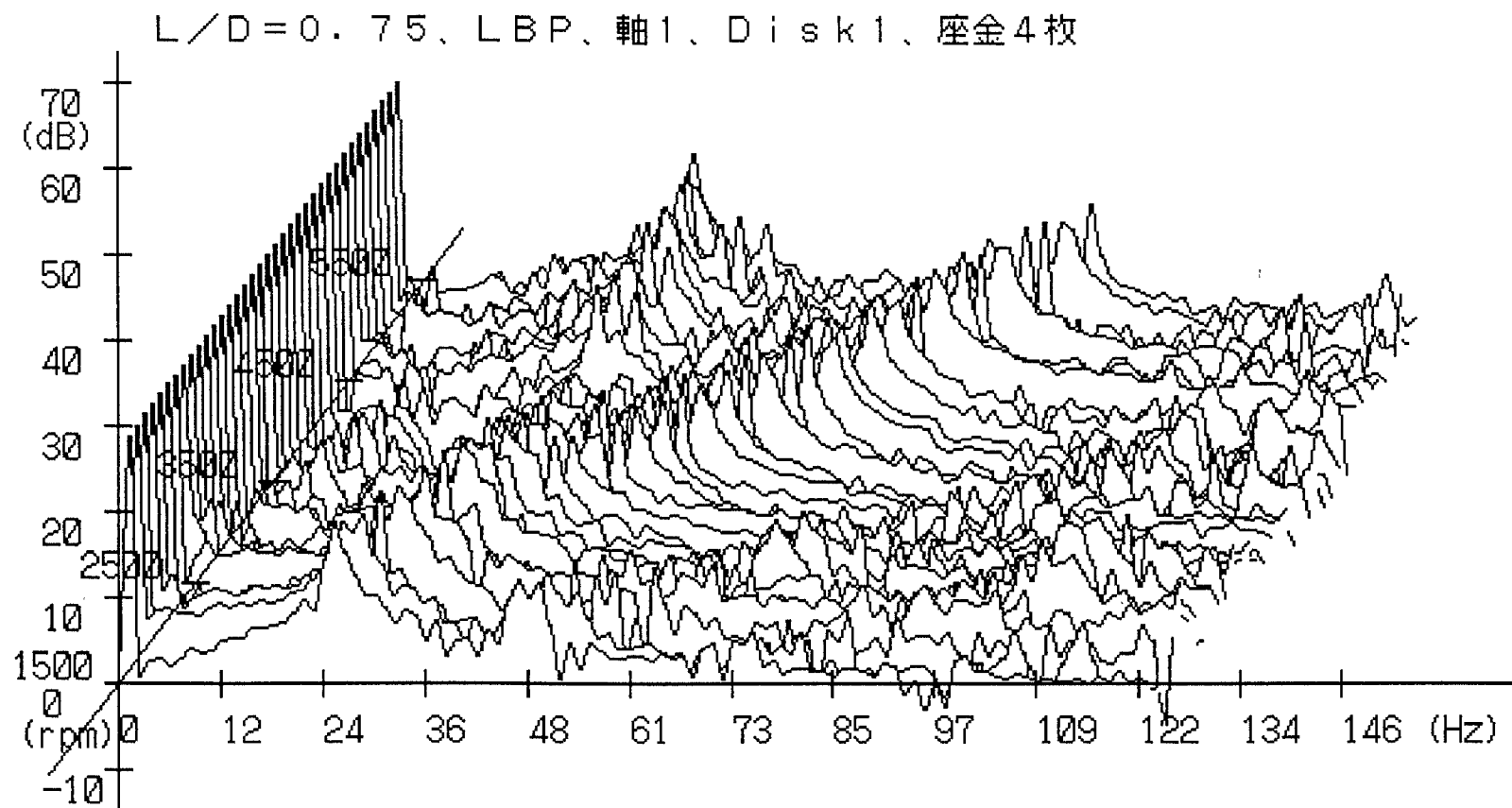
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



回転速度、周波数をパラメータとしたときの振動振幅の解析結果

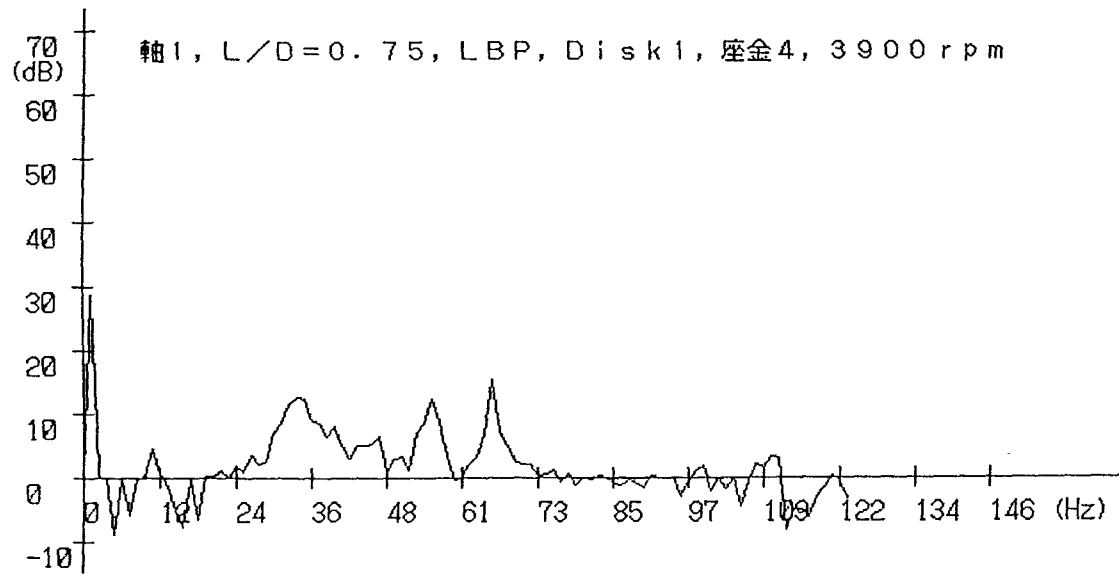
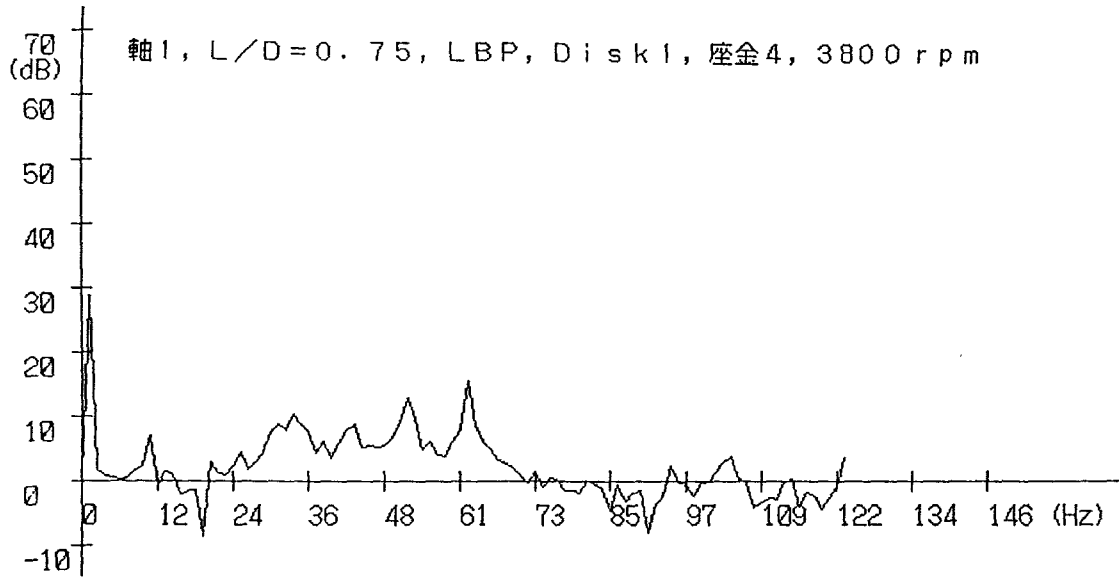


安定限界速度近辺の軸振動振幅

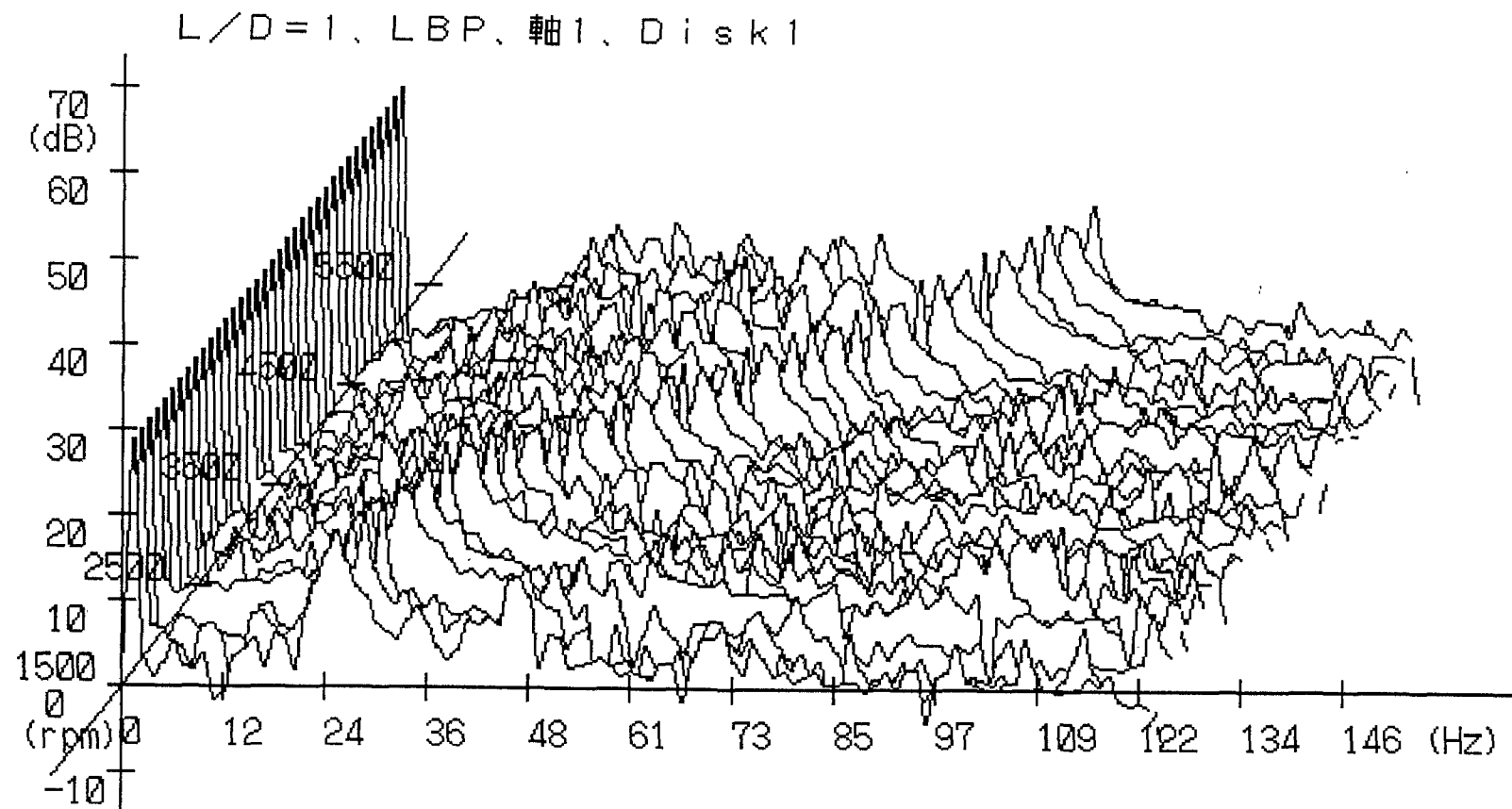


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

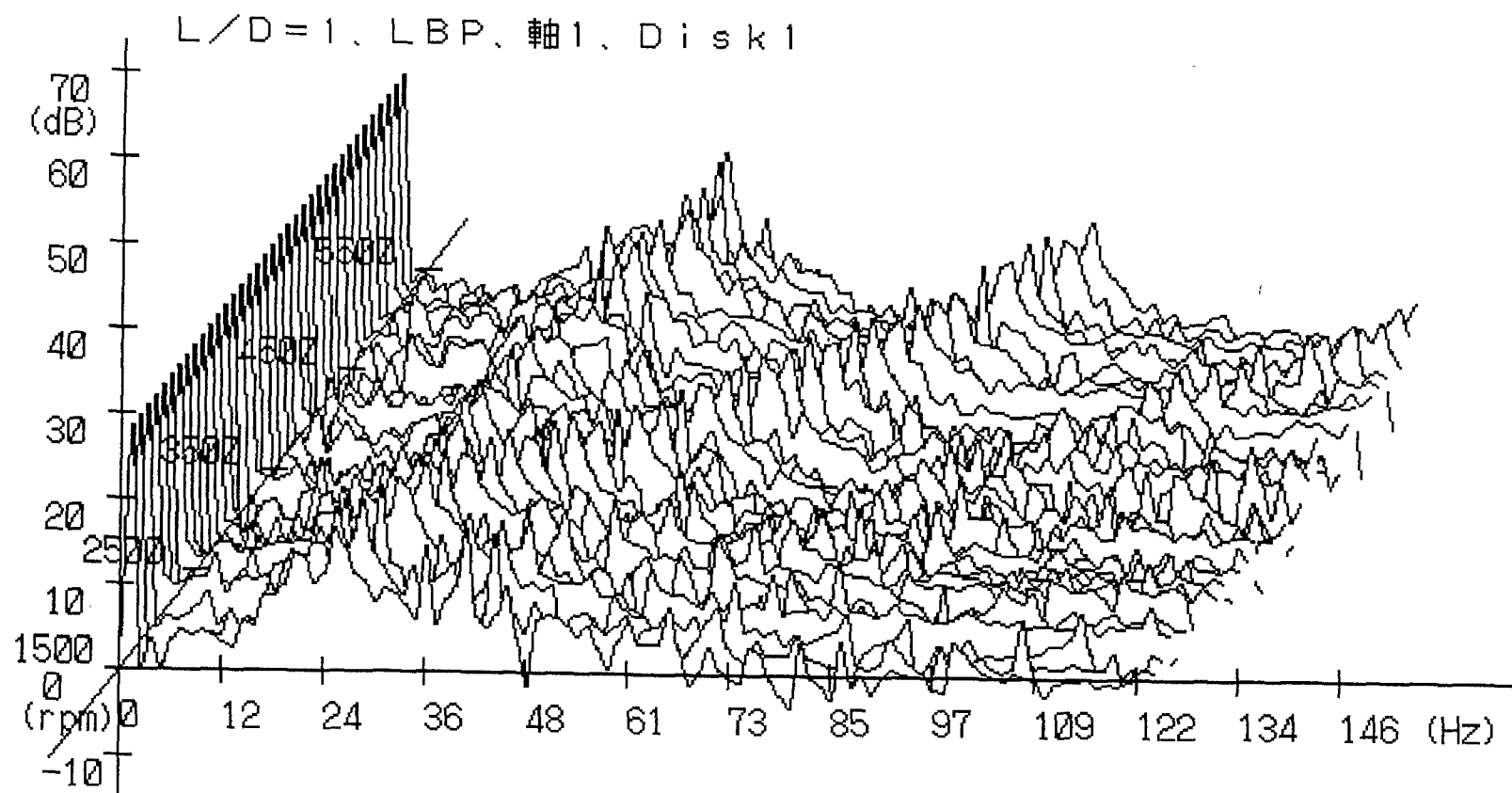




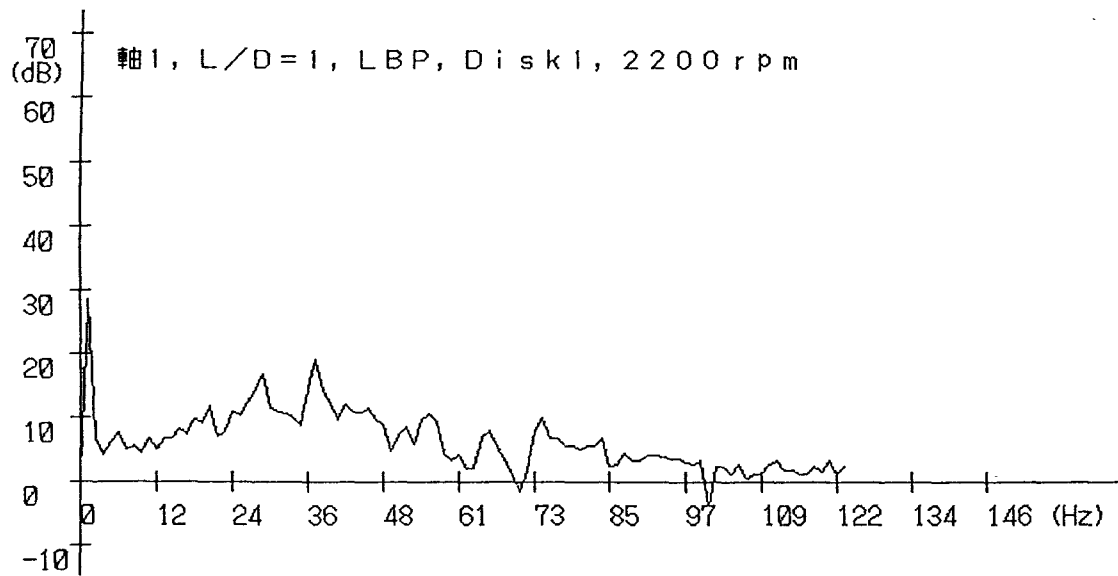
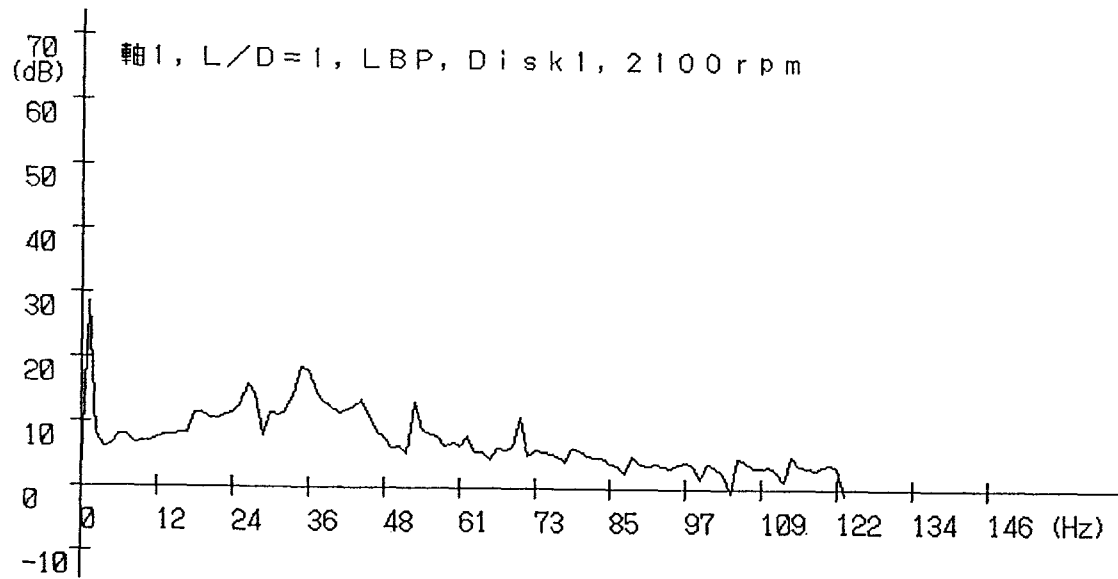
安定限界速度近辺の軸振動振幅



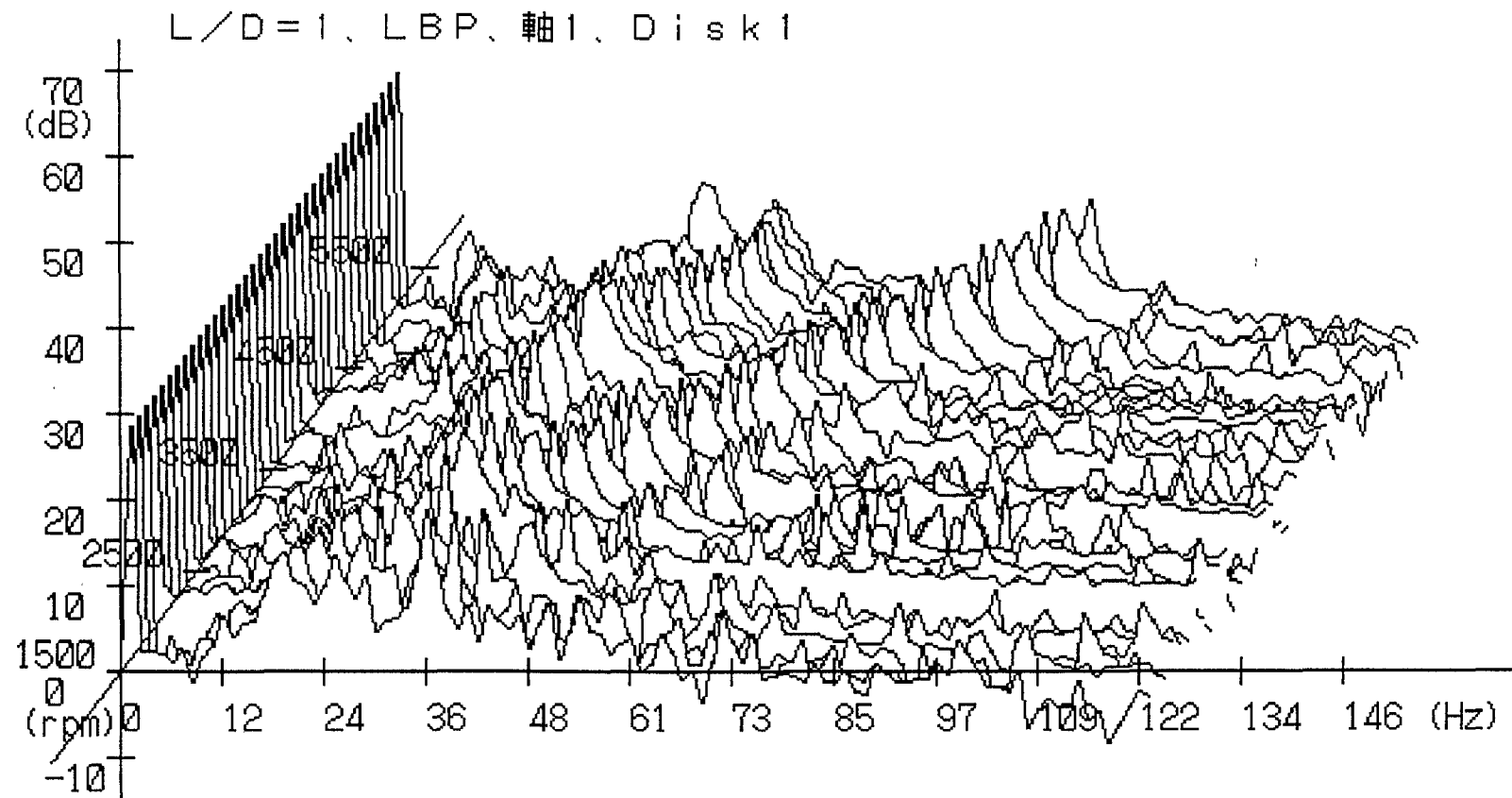
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



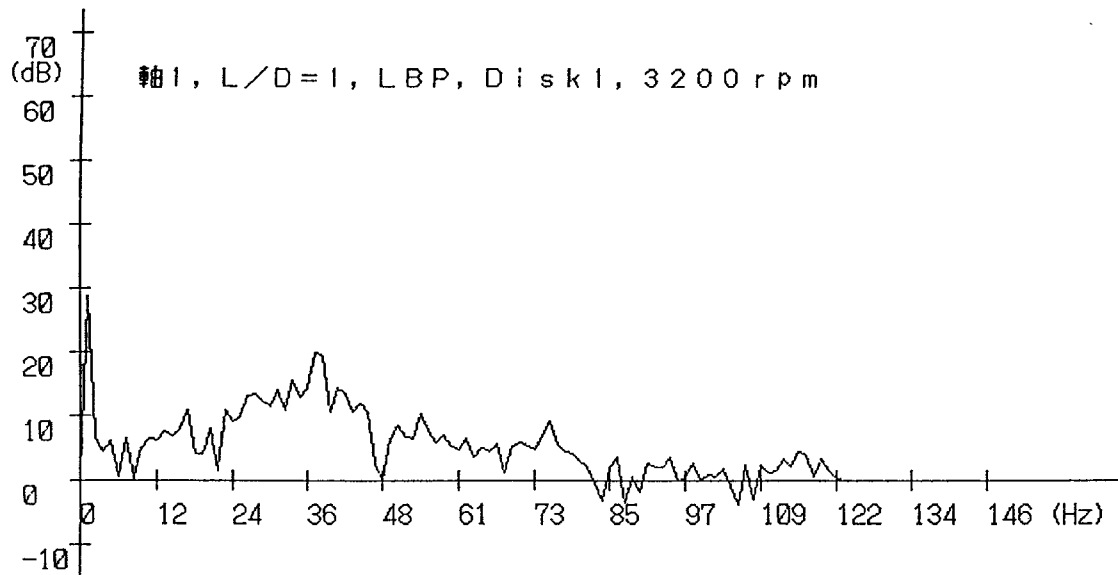
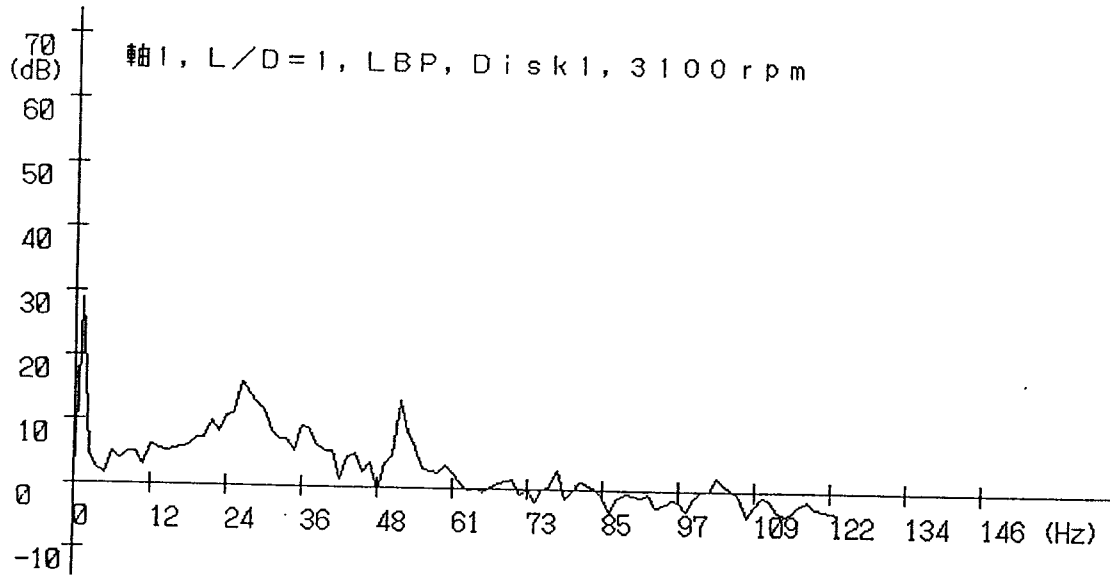
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



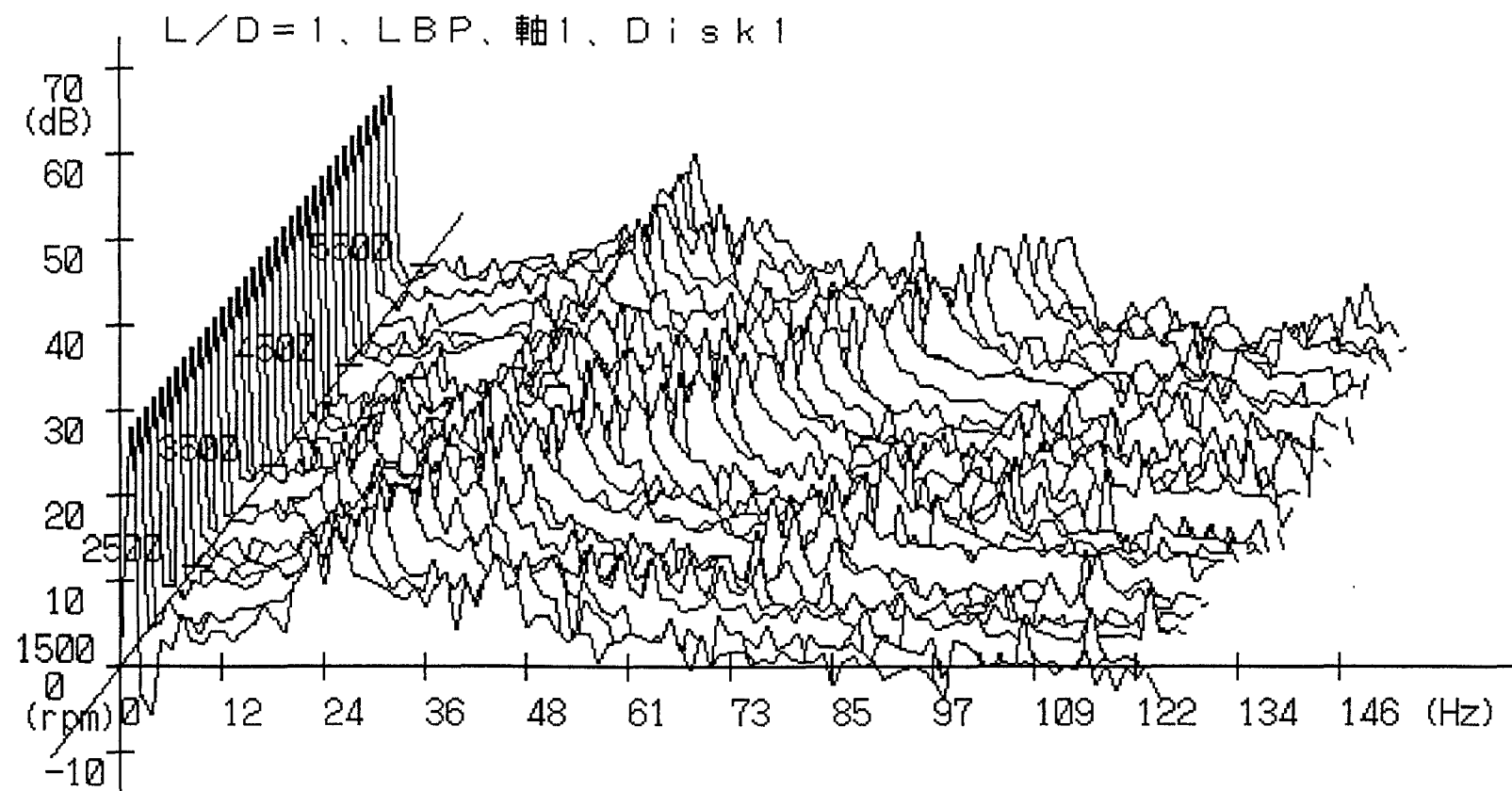
安定限界速度近辺の軸振動振幅



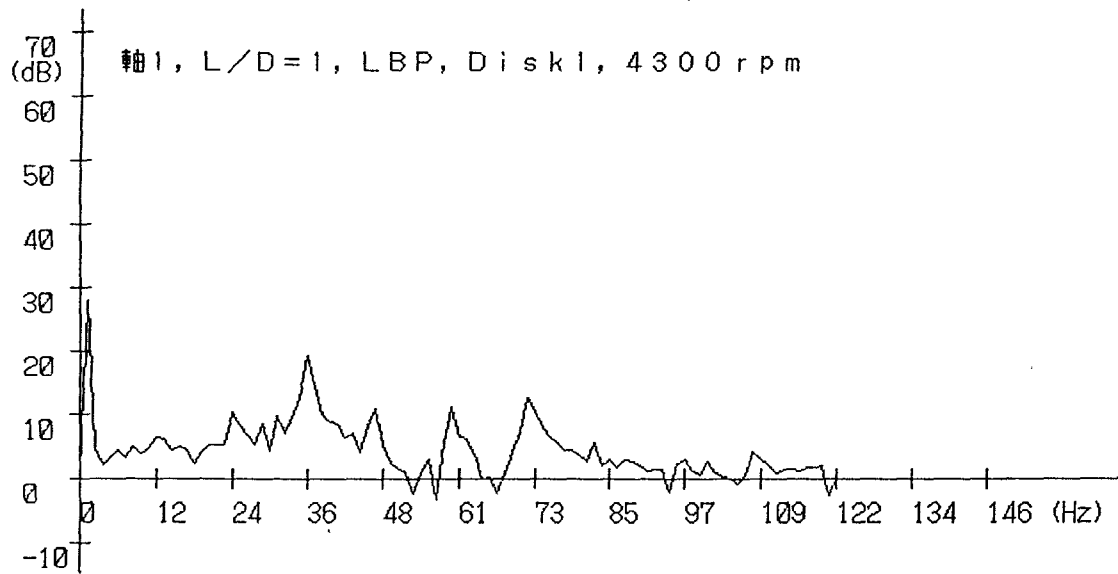
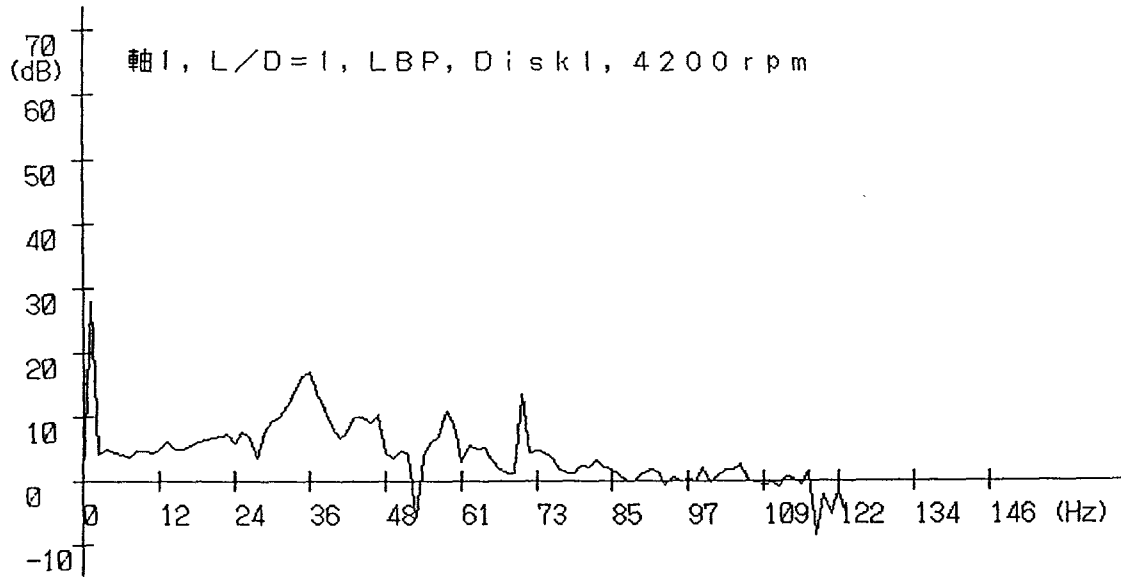
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



安定限界速度近辺の軸振動振幅

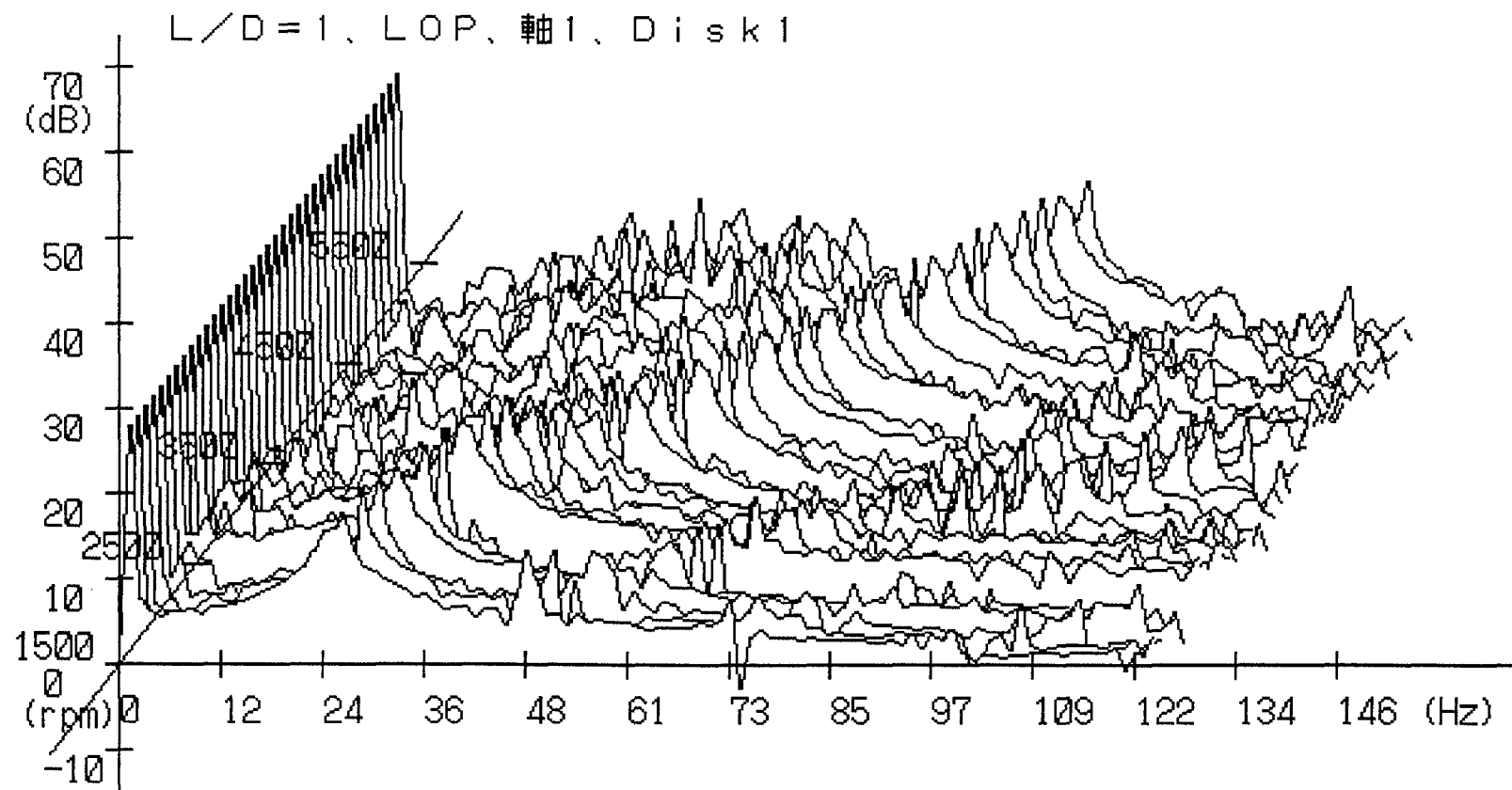


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

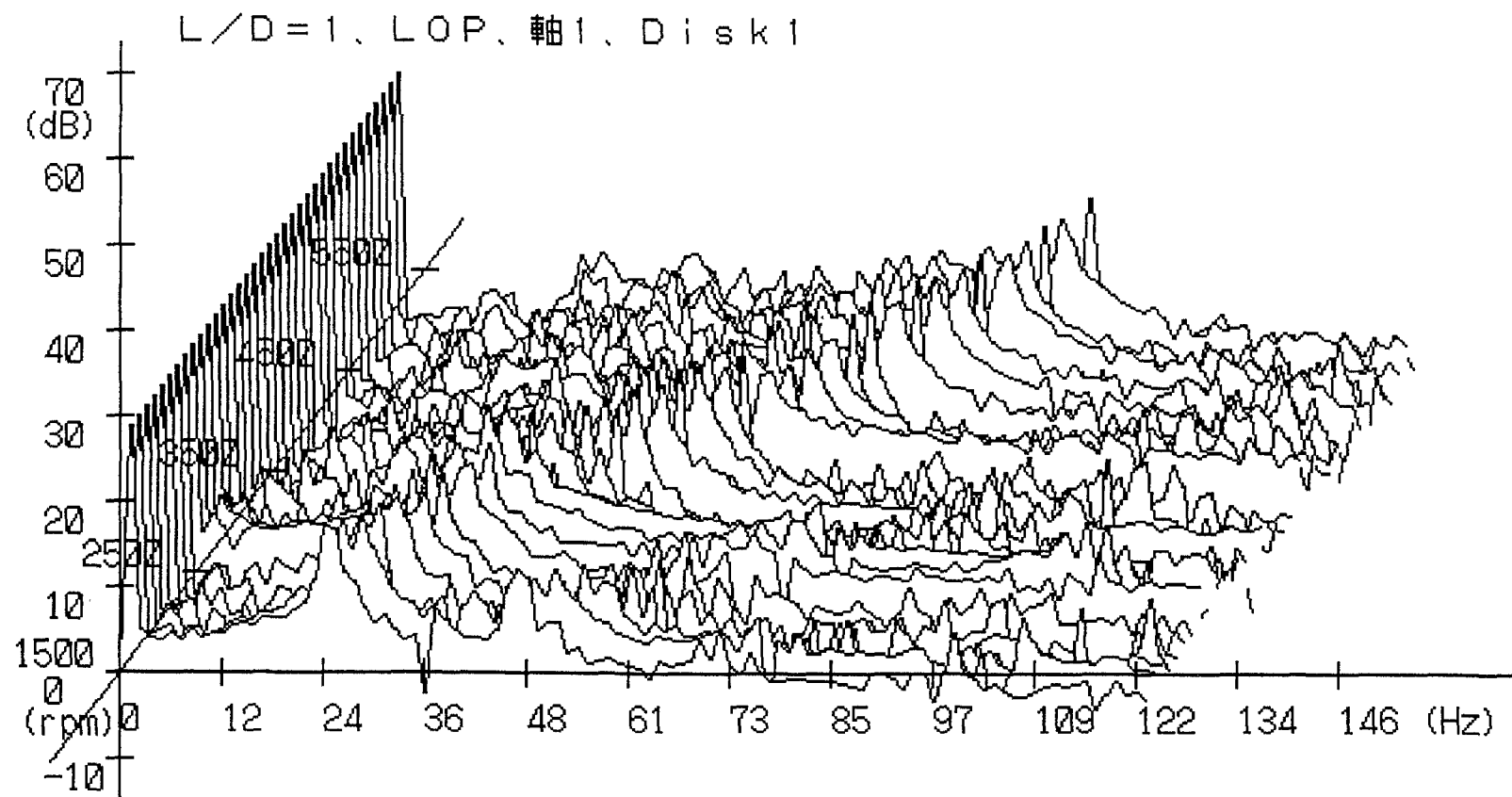


安定限界速度近辺の軸振動振幅

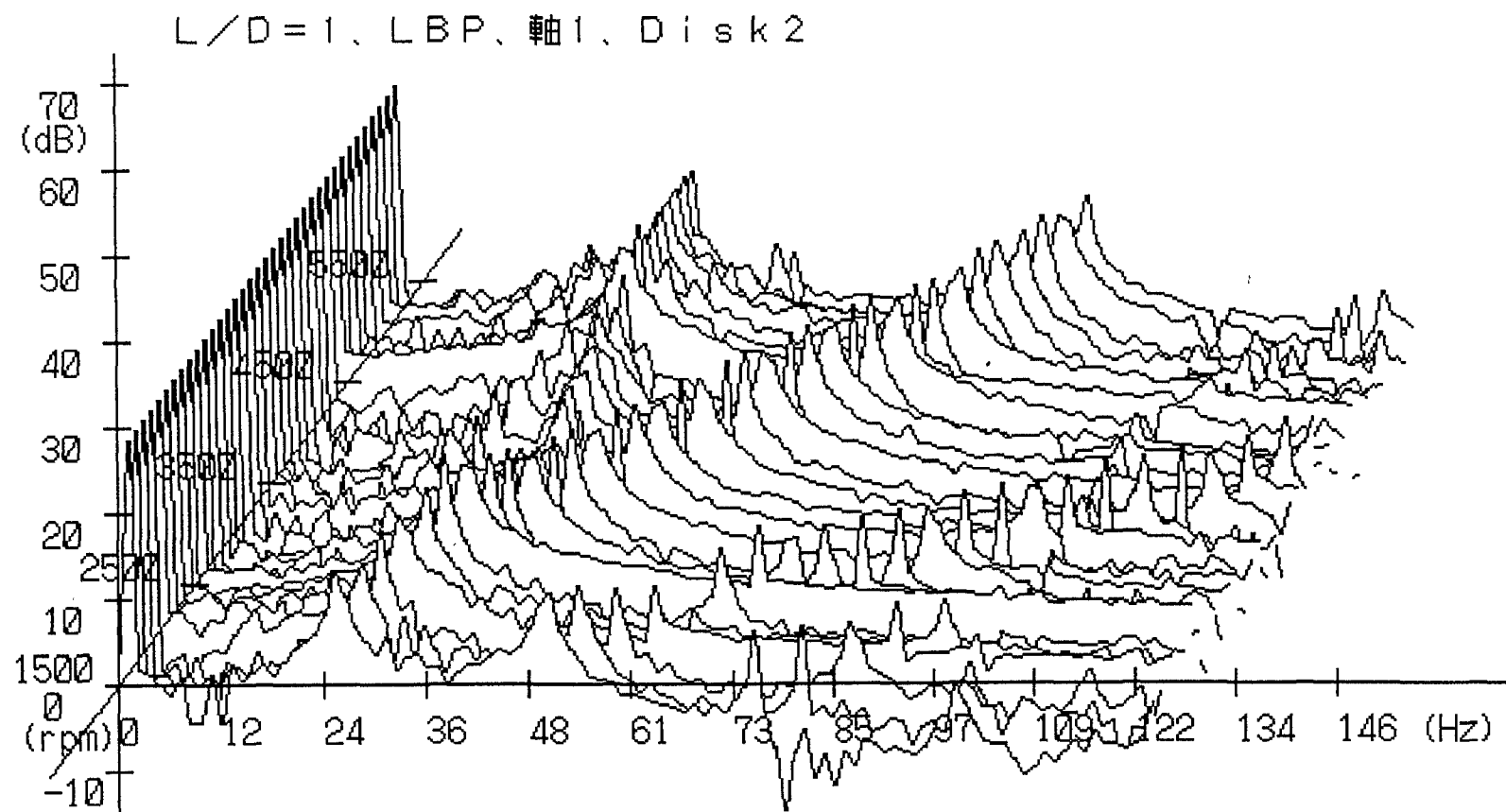




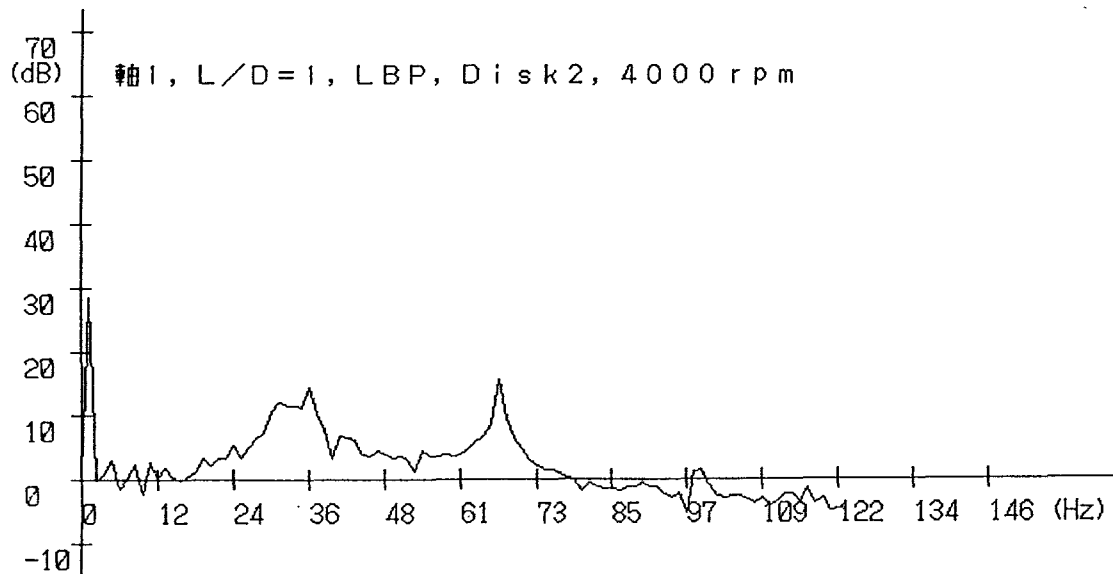
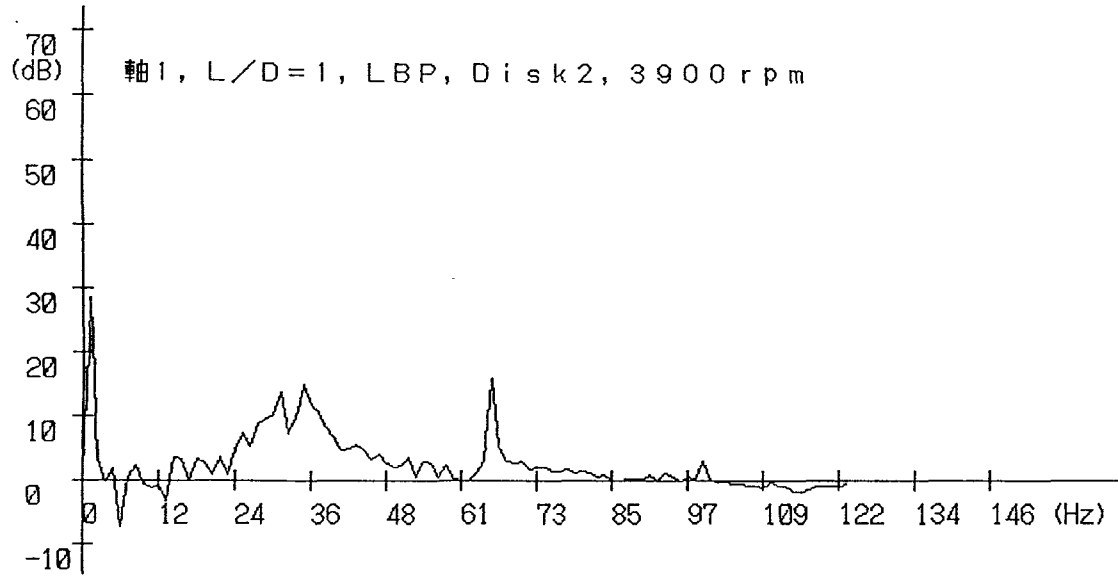
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



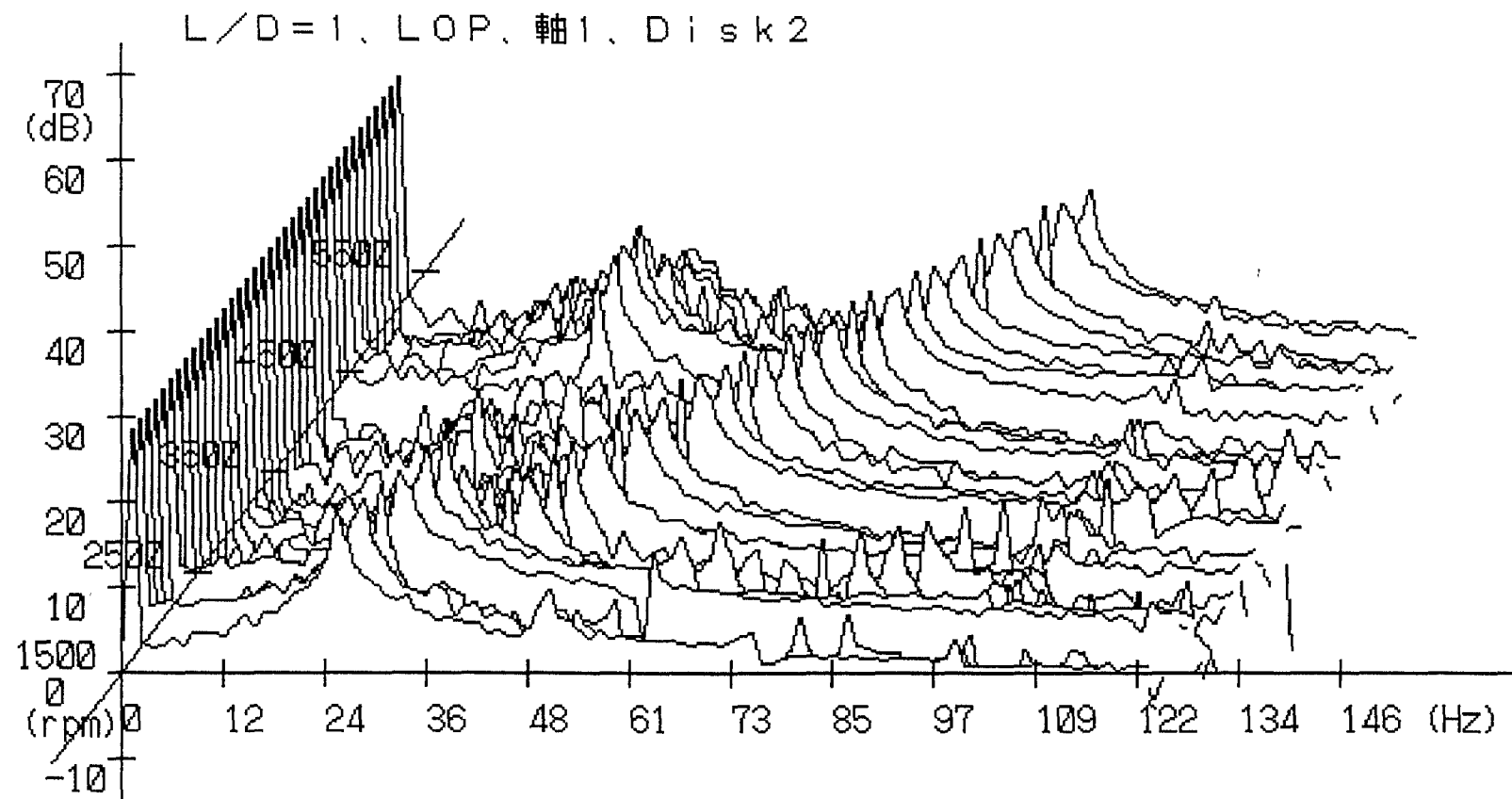
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



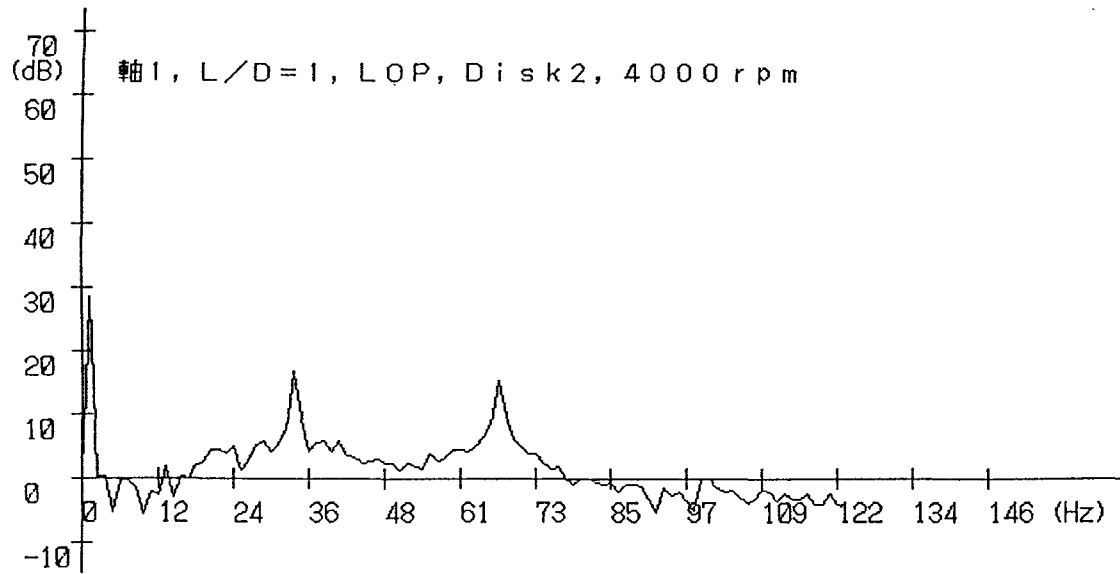
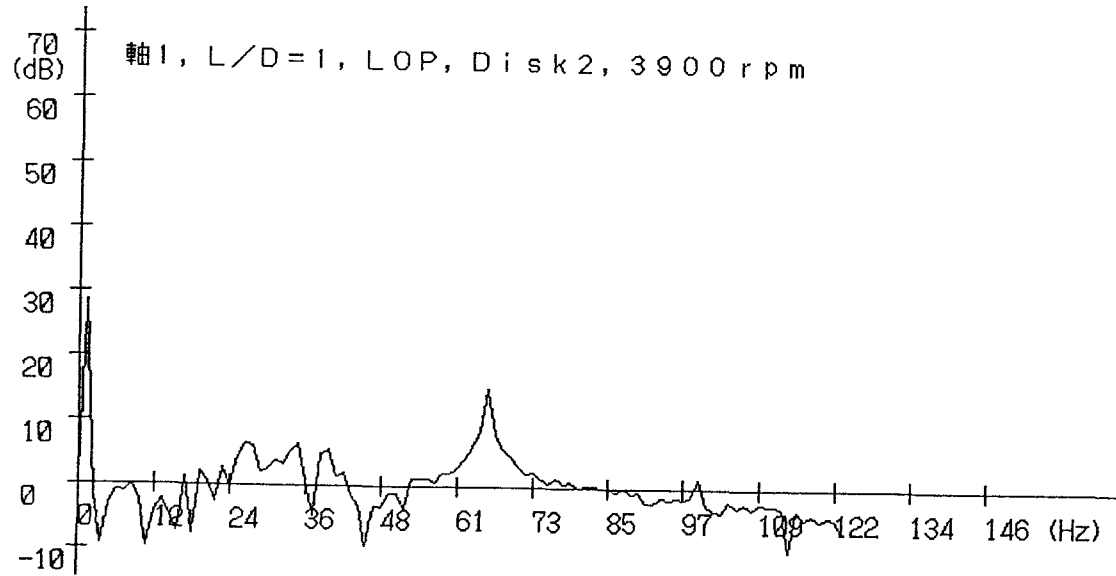
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



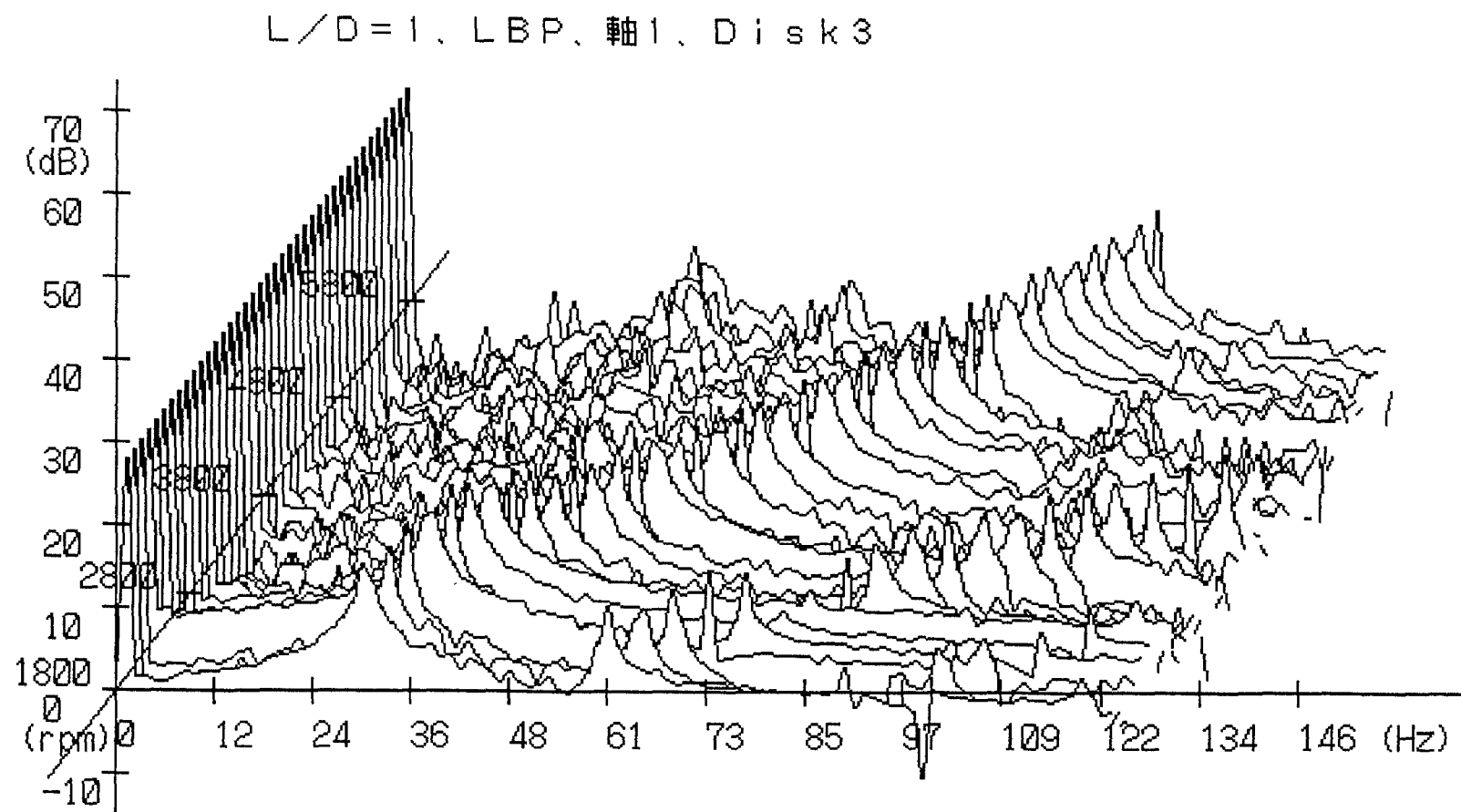
安定限界速度近辺の軸振動振幅



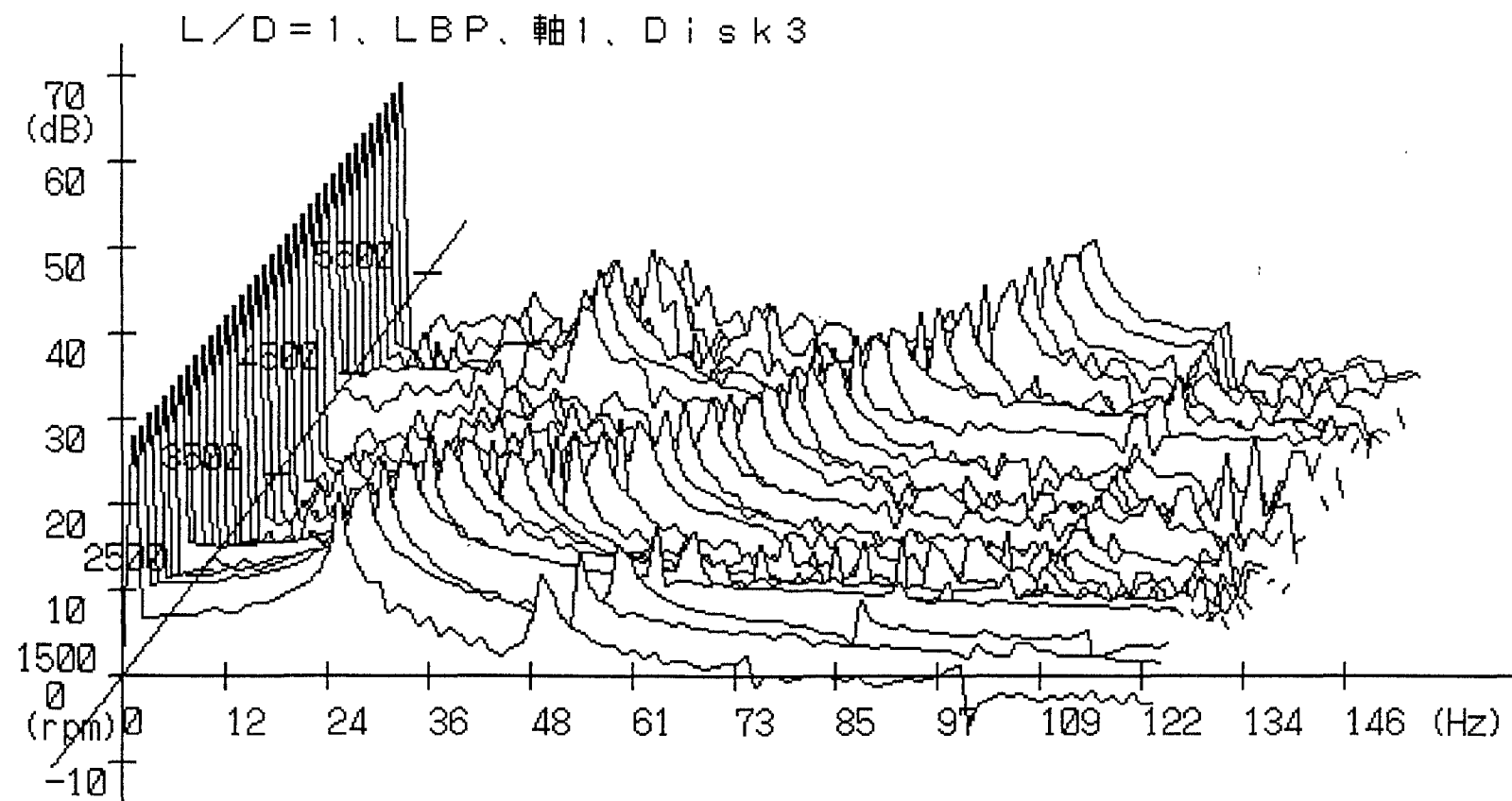
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



安定限界速度近辺の軸振動振幅

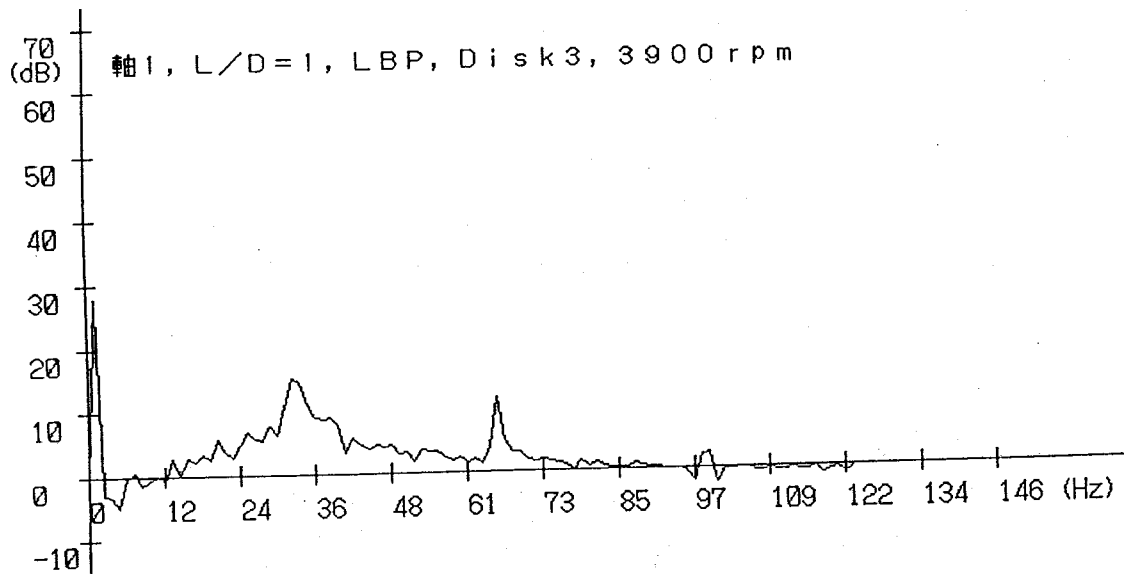
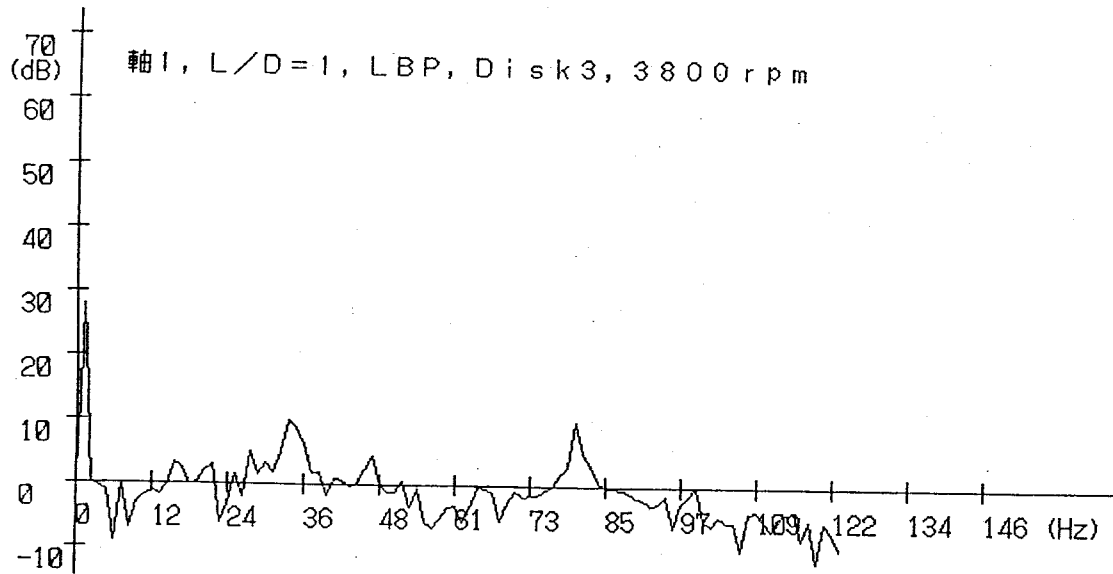


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

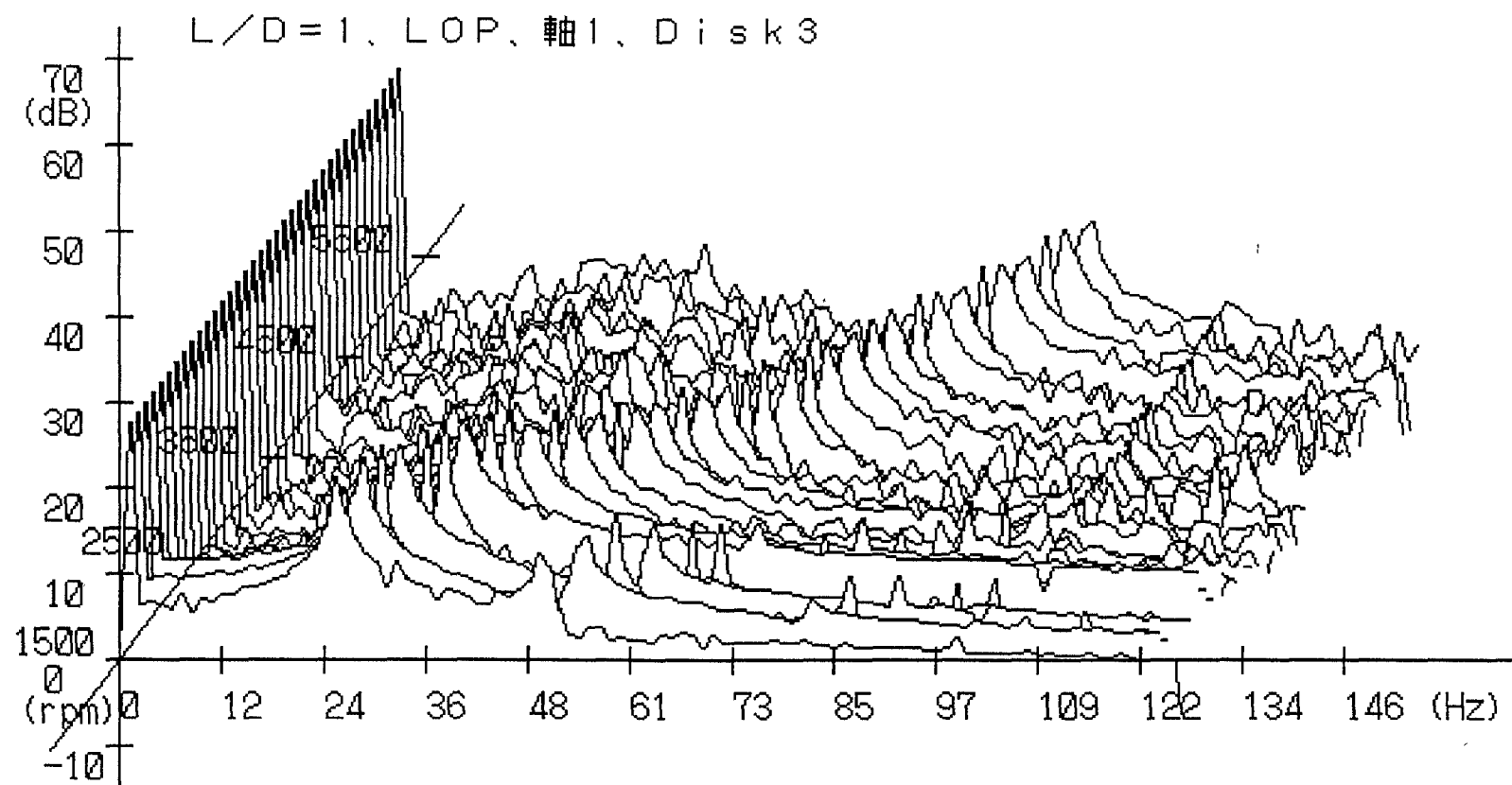


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

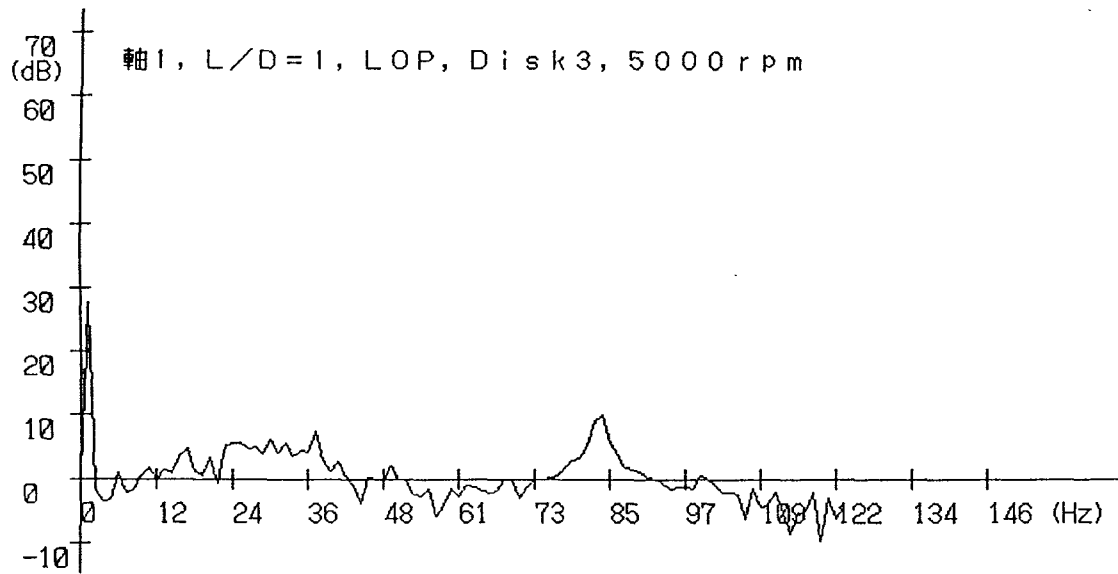
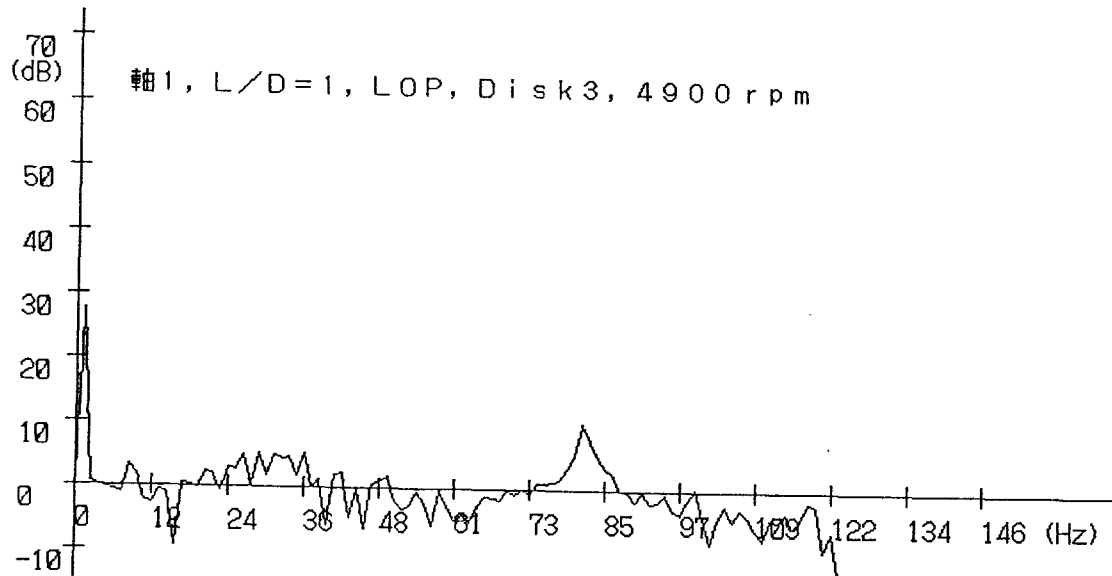




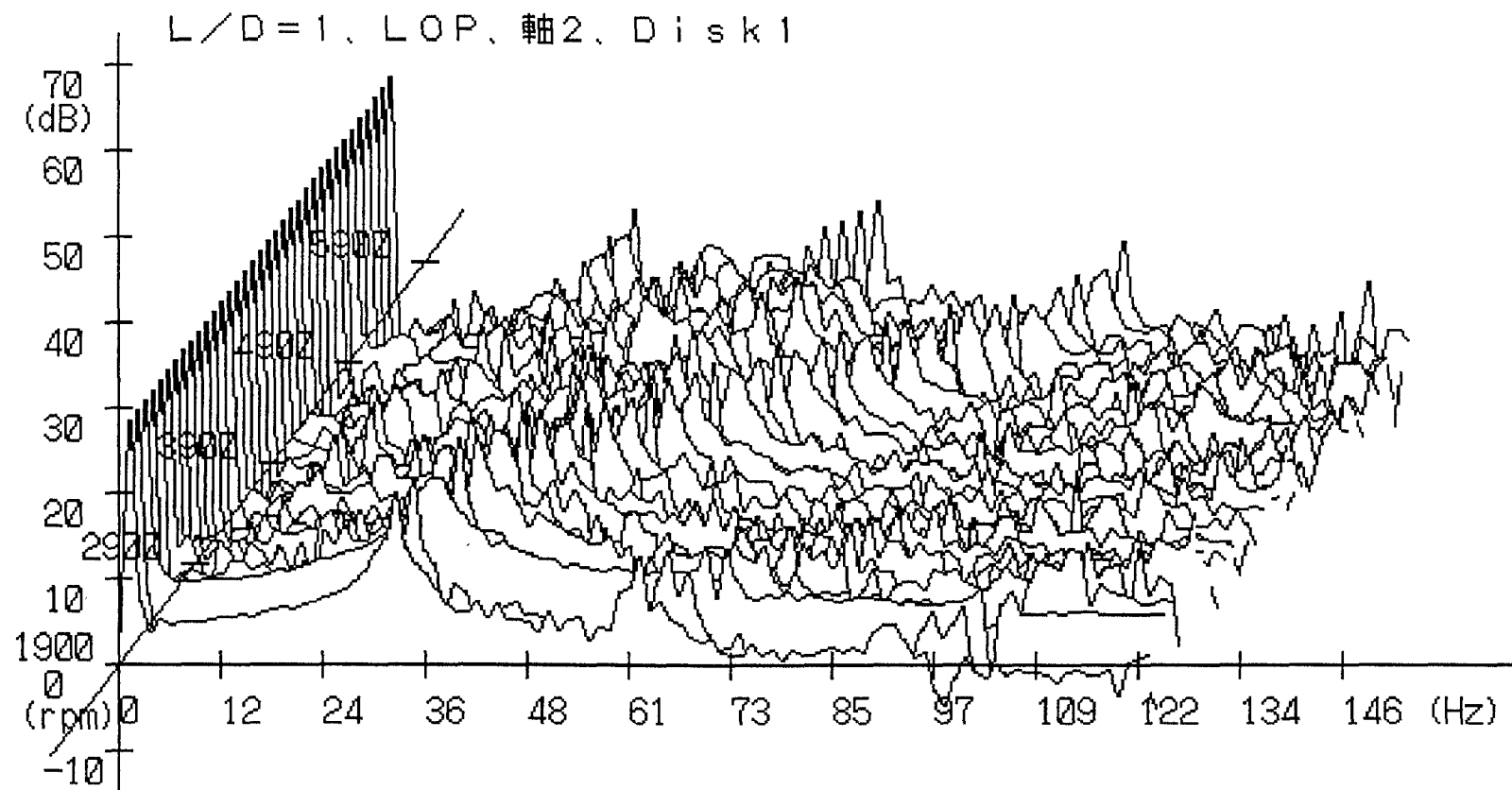
安定限界速度近辺の軸振動振幅



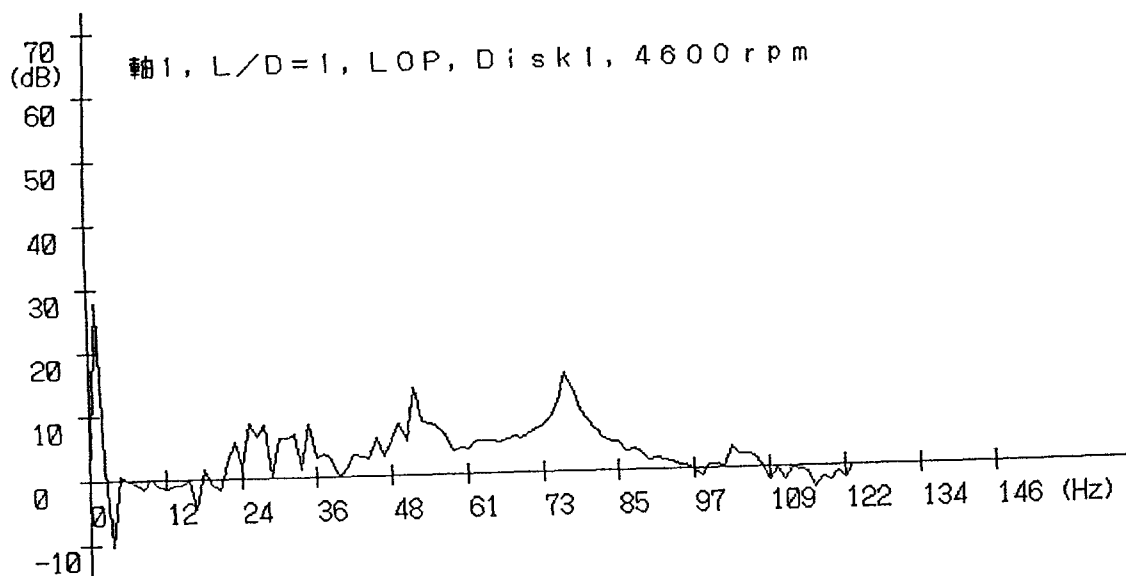
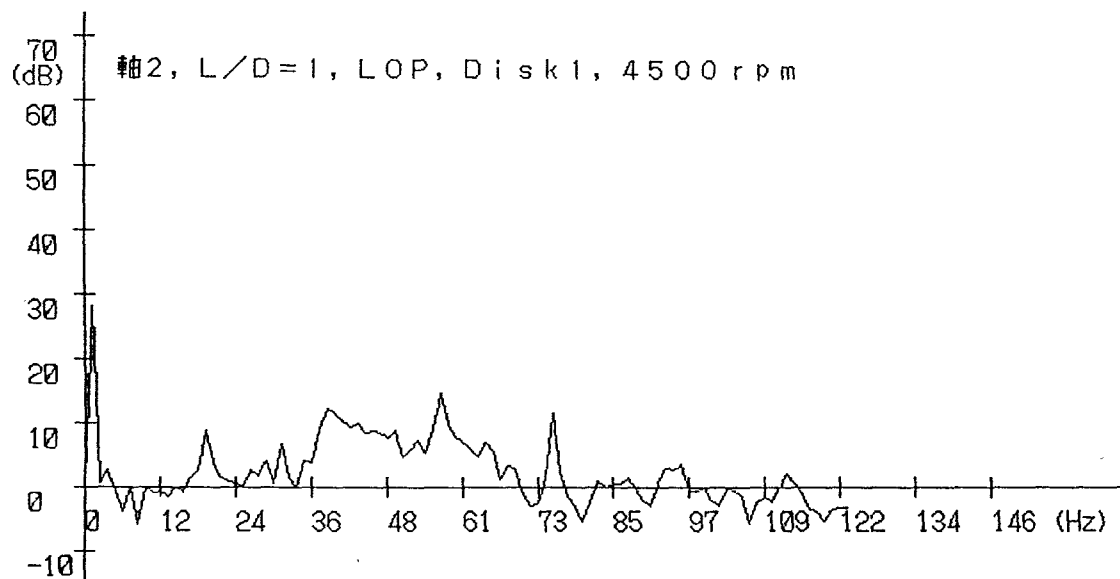
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



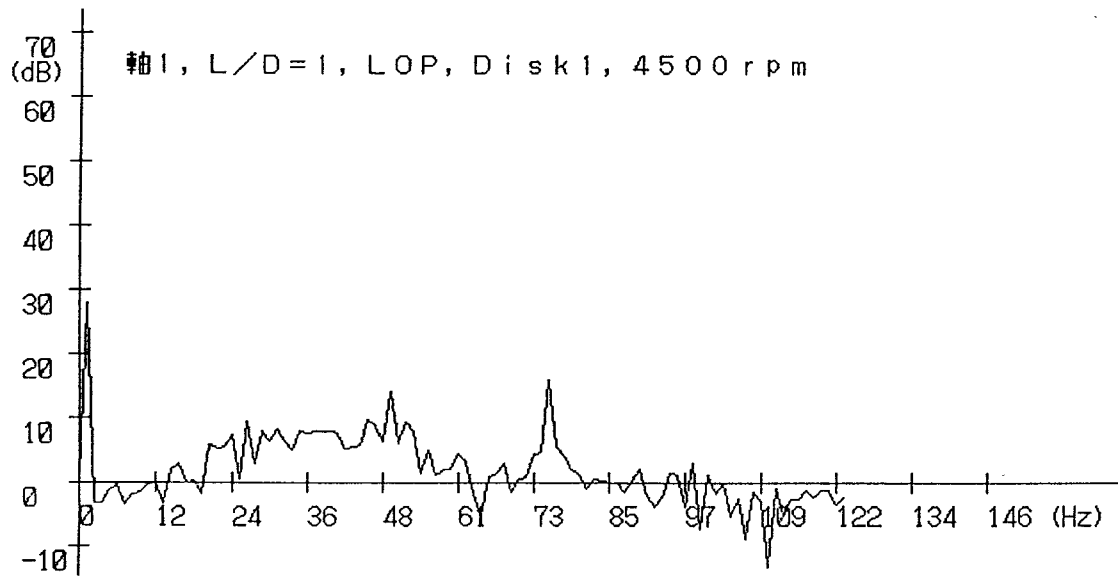
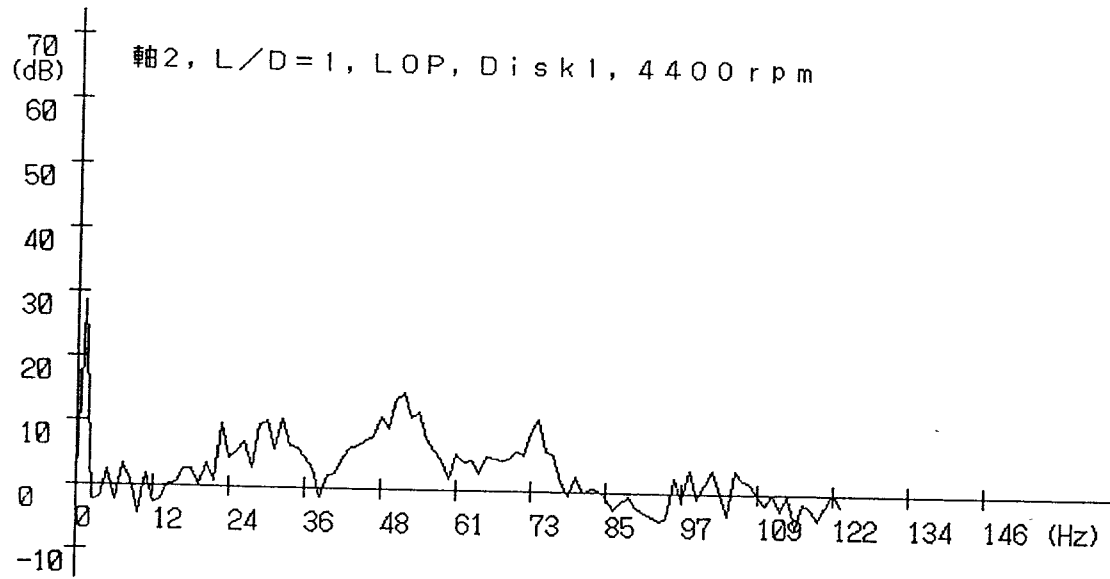
安定限界速度近辺の軸振動振幅



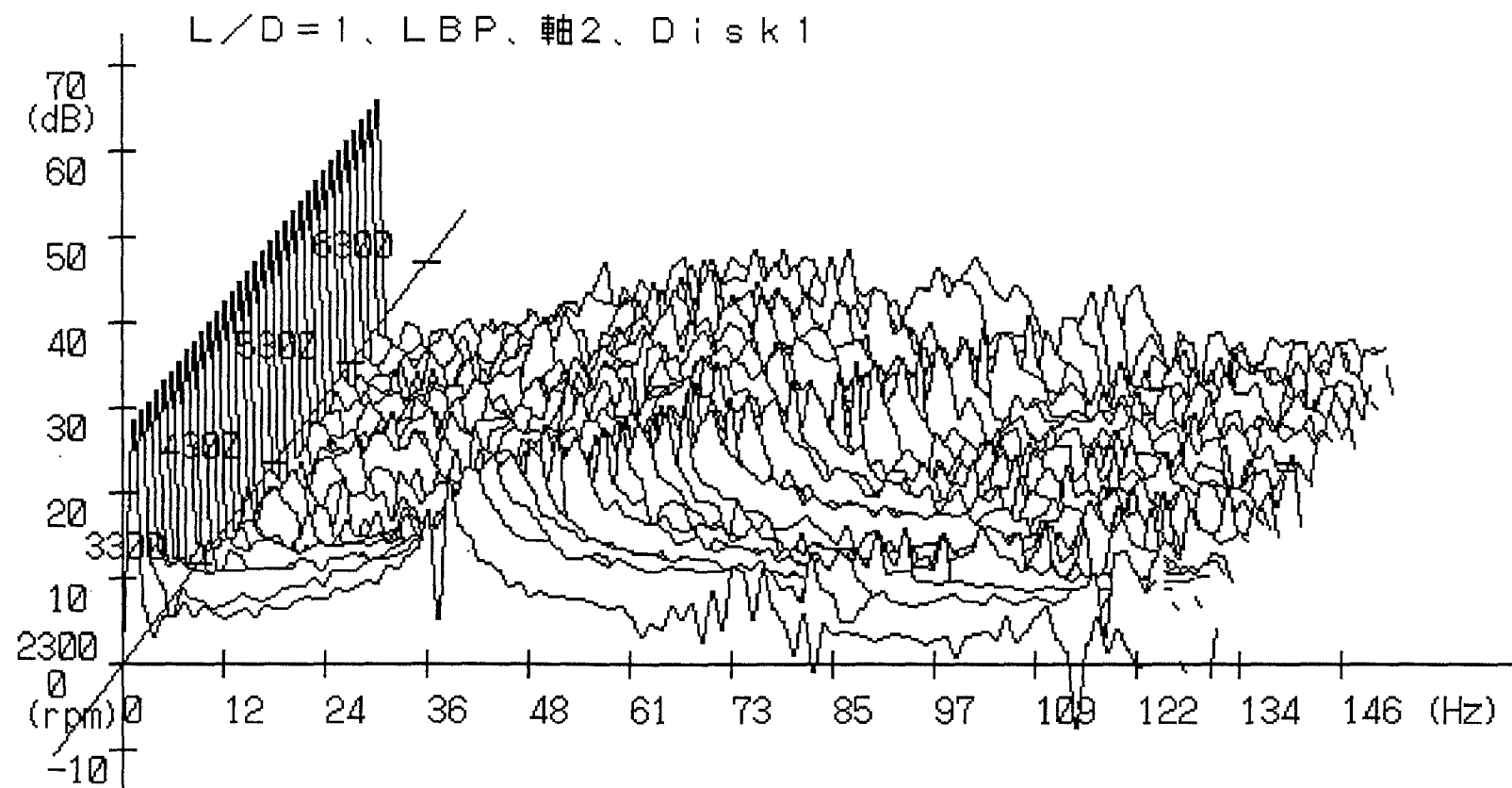
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



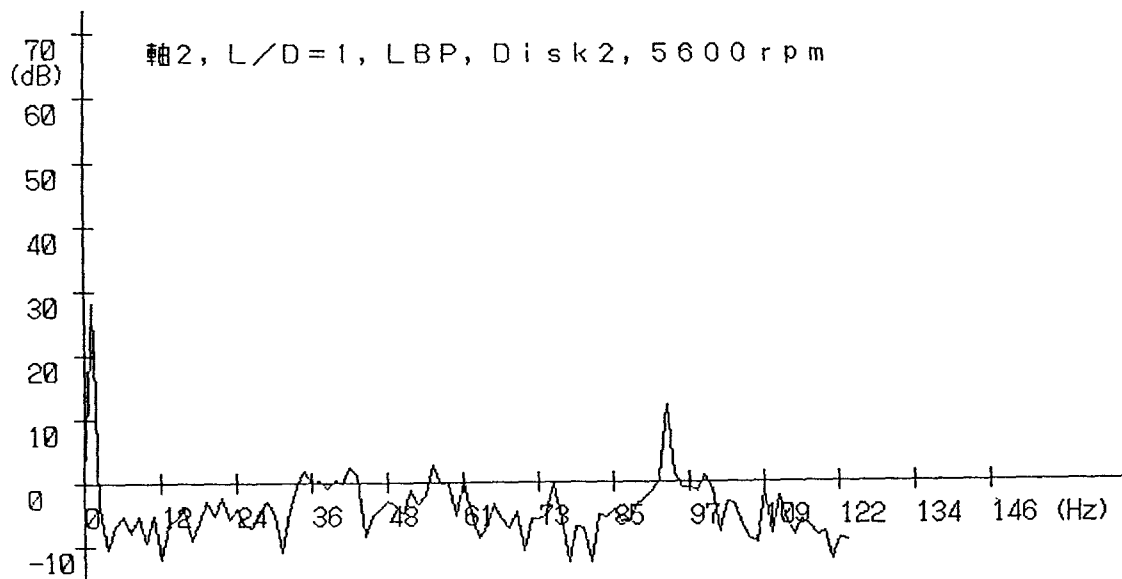
安定限界速度近辺の軸振動振幅



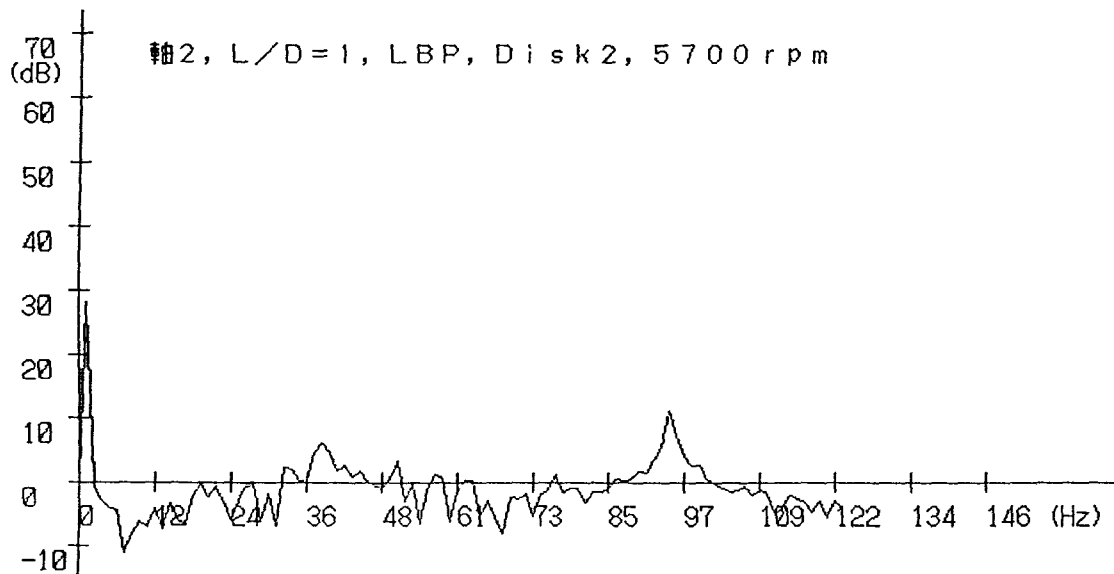
安定限界速度近辺の軸振動振幅



回転速度、周波数をパラメータとしたときの振動振幅の解析結果

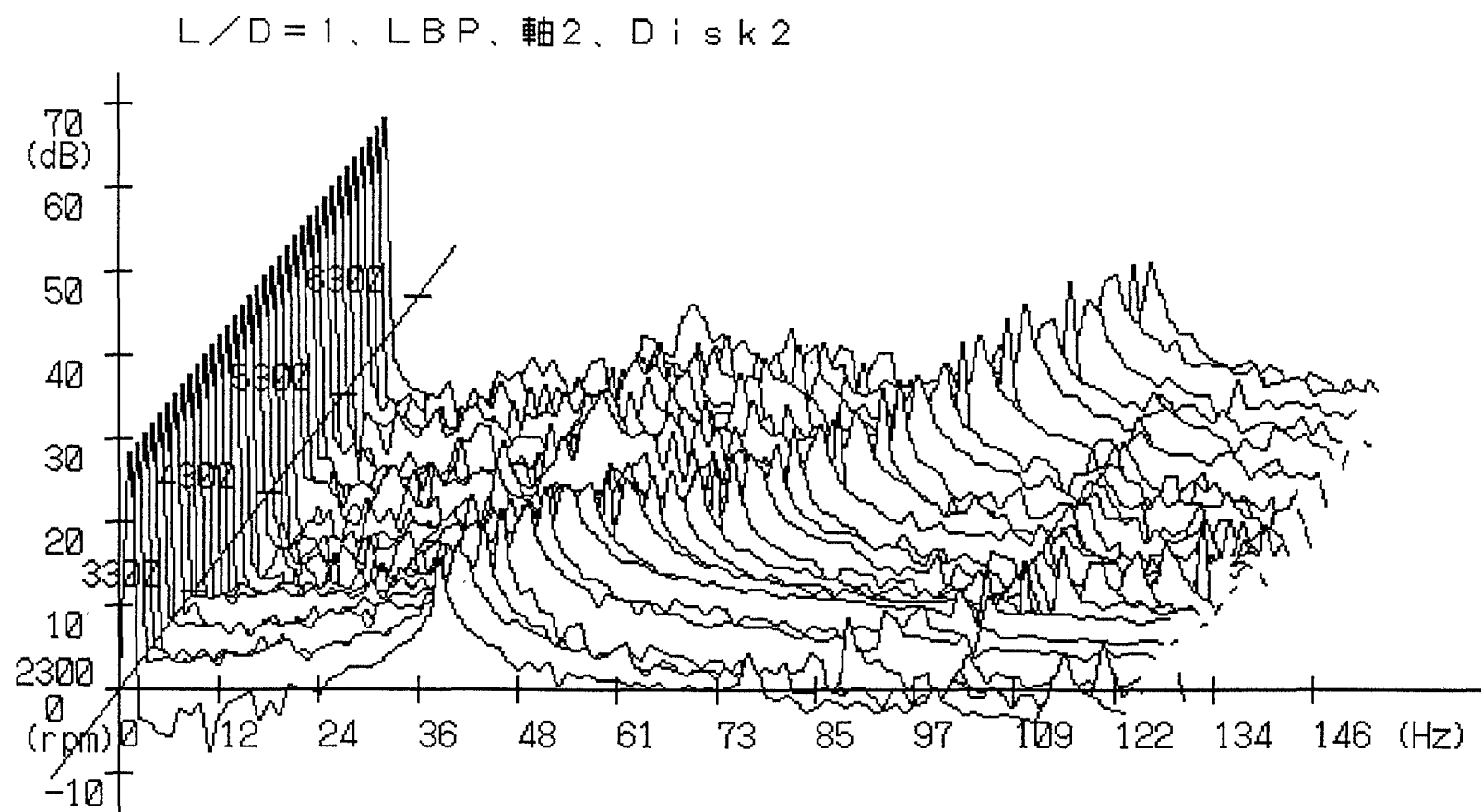


L

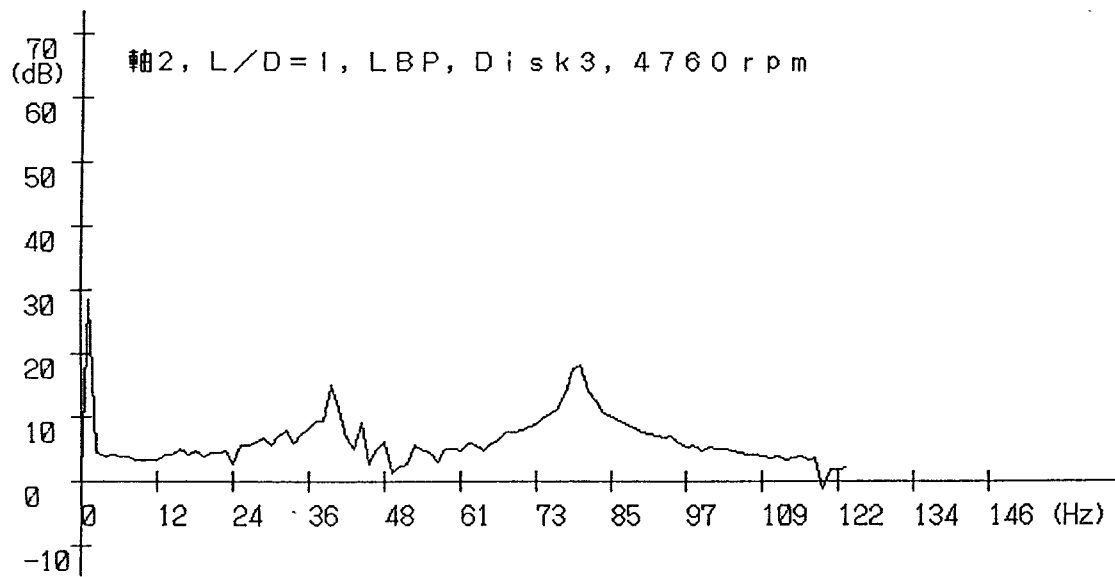
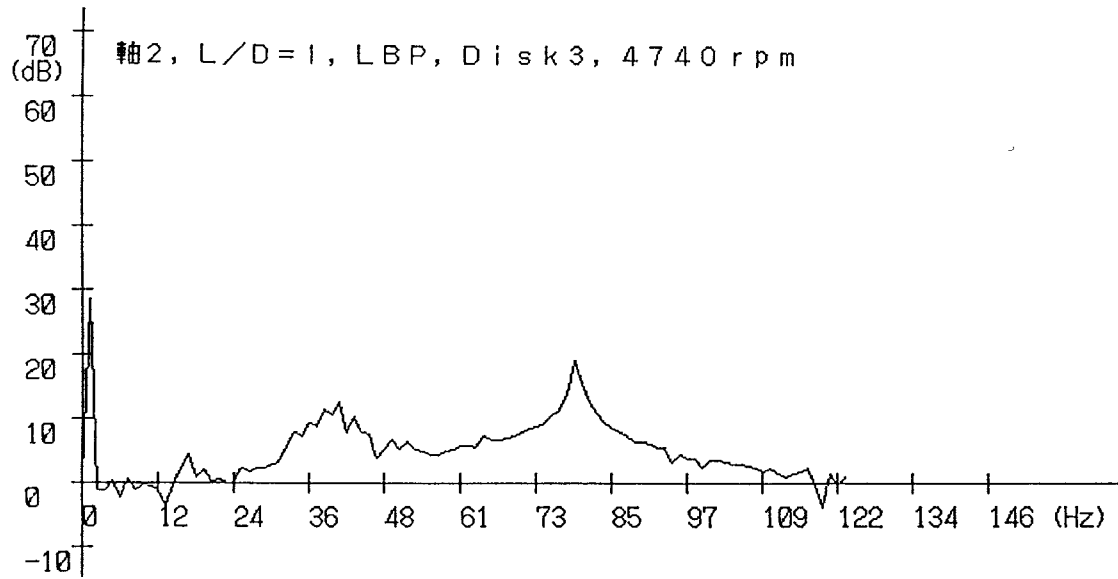


安定限界速度近辺の軸振動振幅

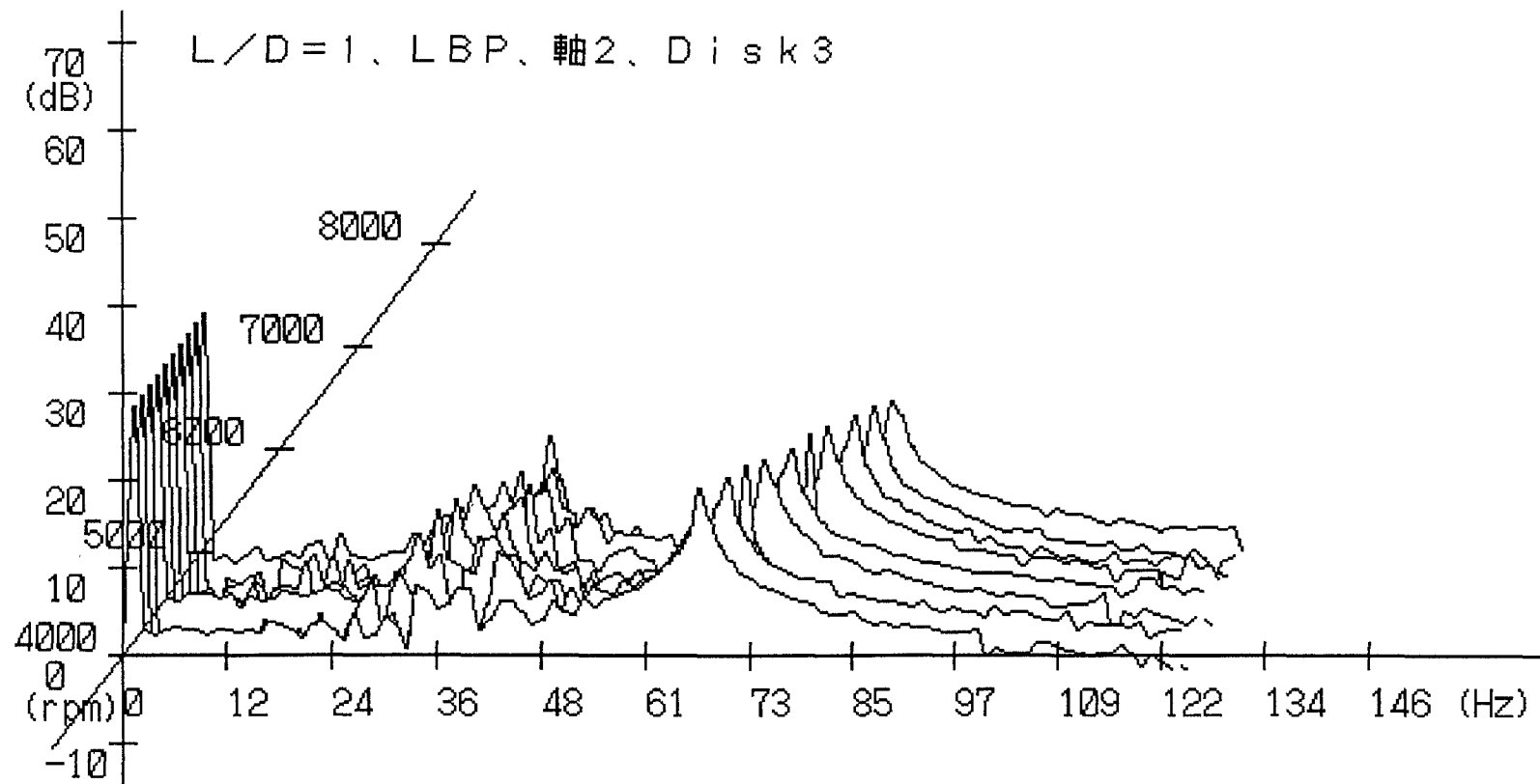




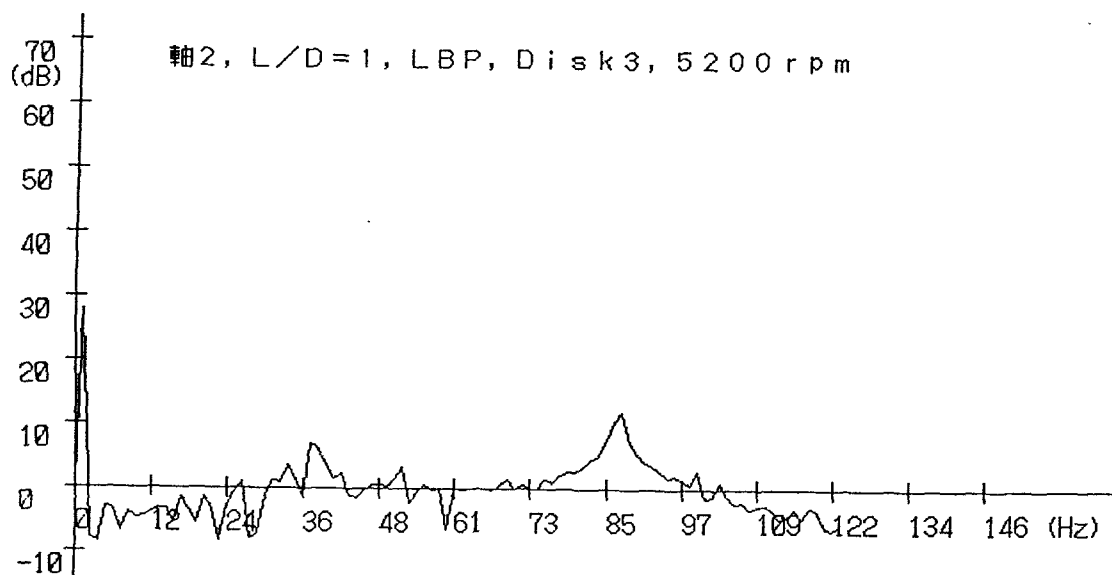
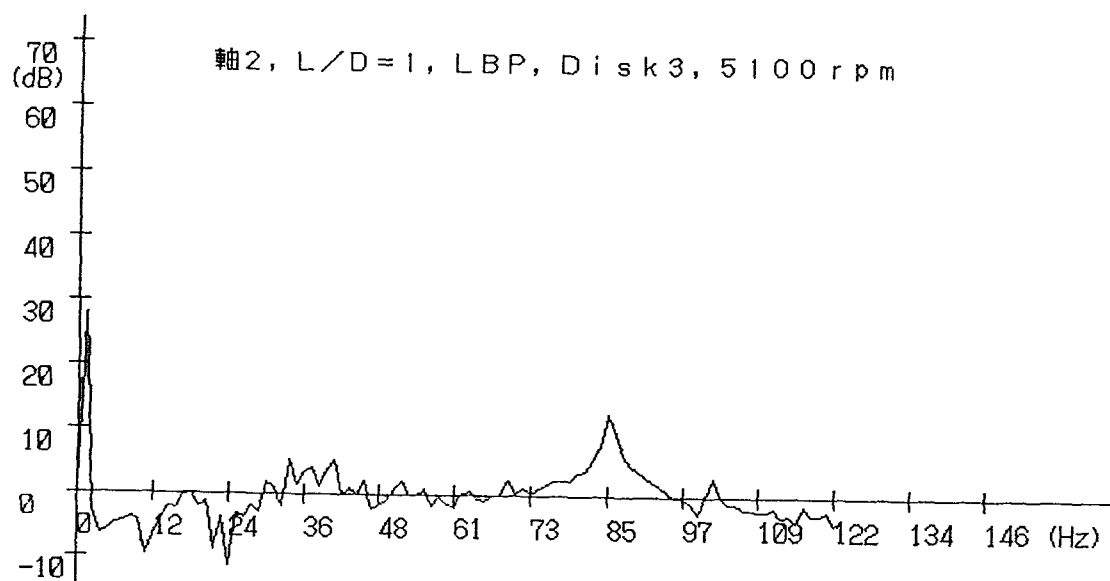
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



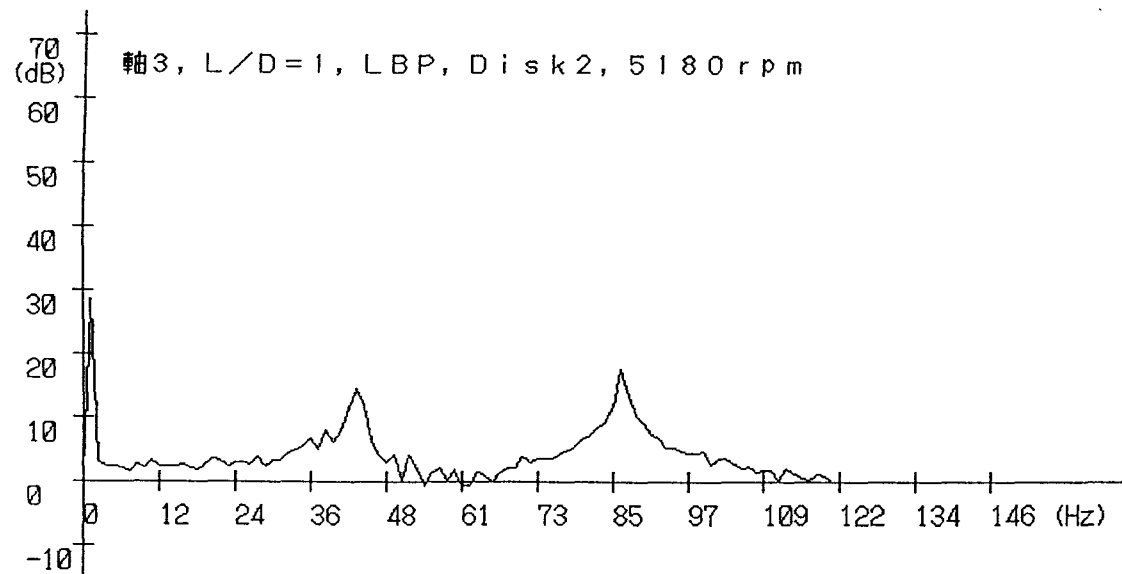
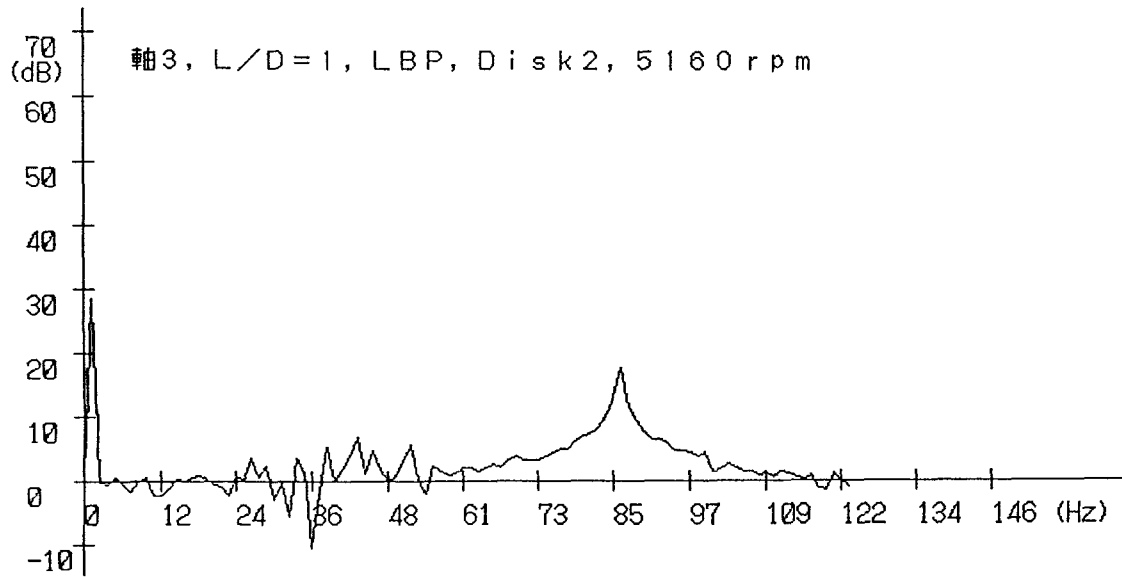
安定限界速度近辺の軸振動振幅



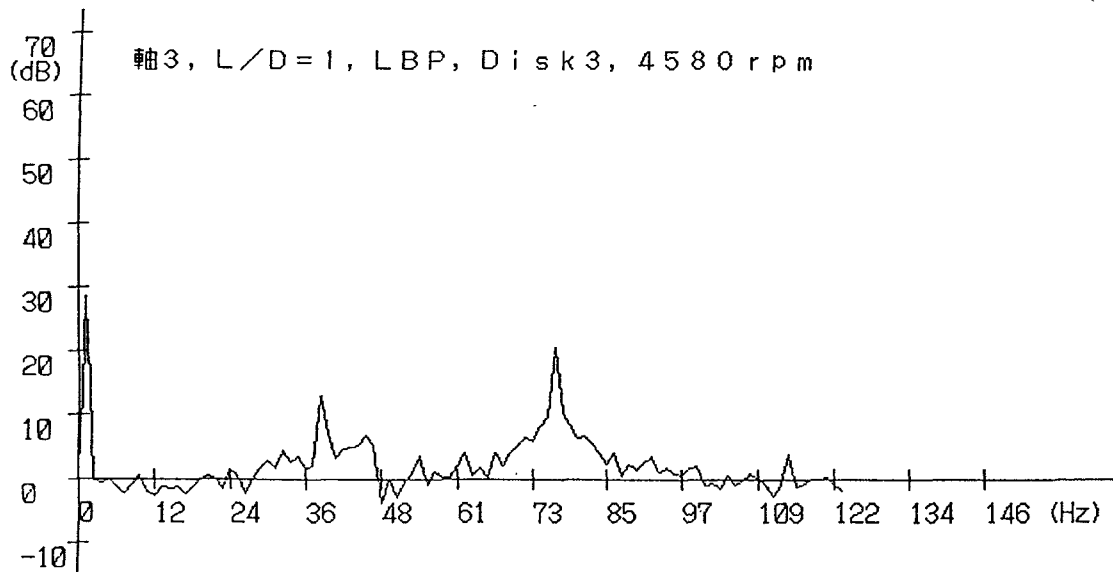
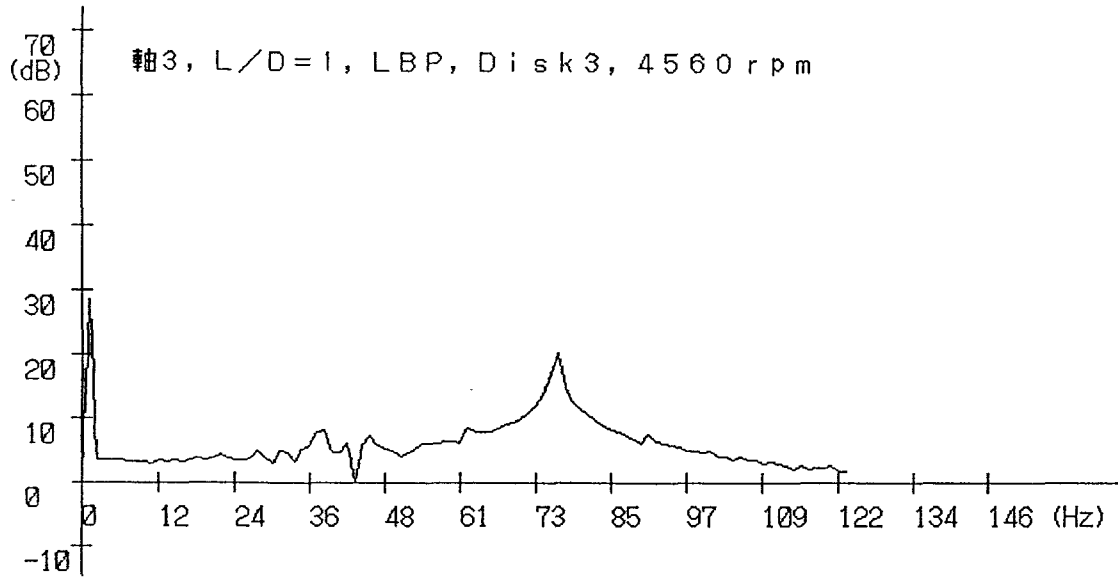
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



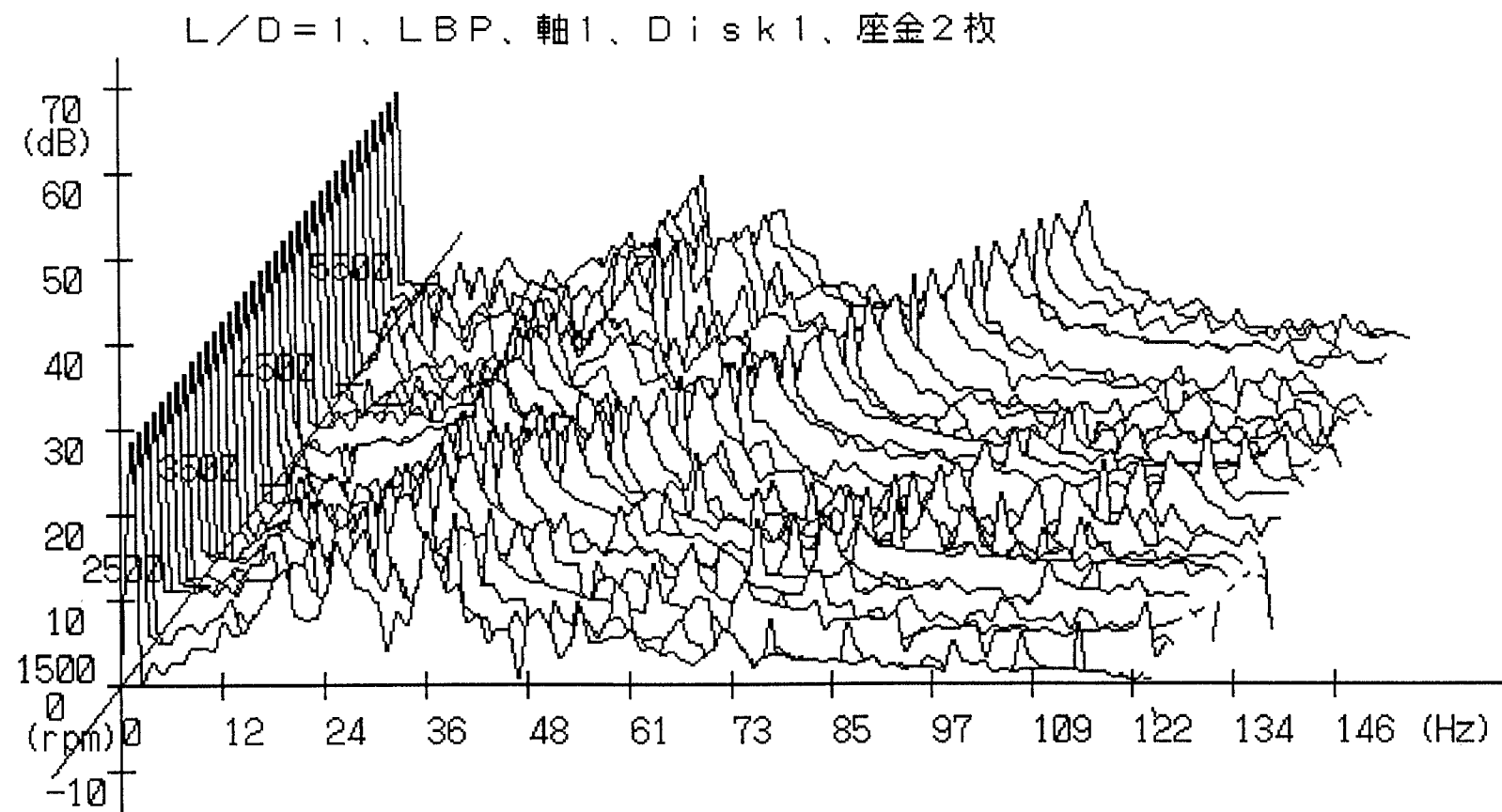
安定限界速度近辺の軸振動振幅



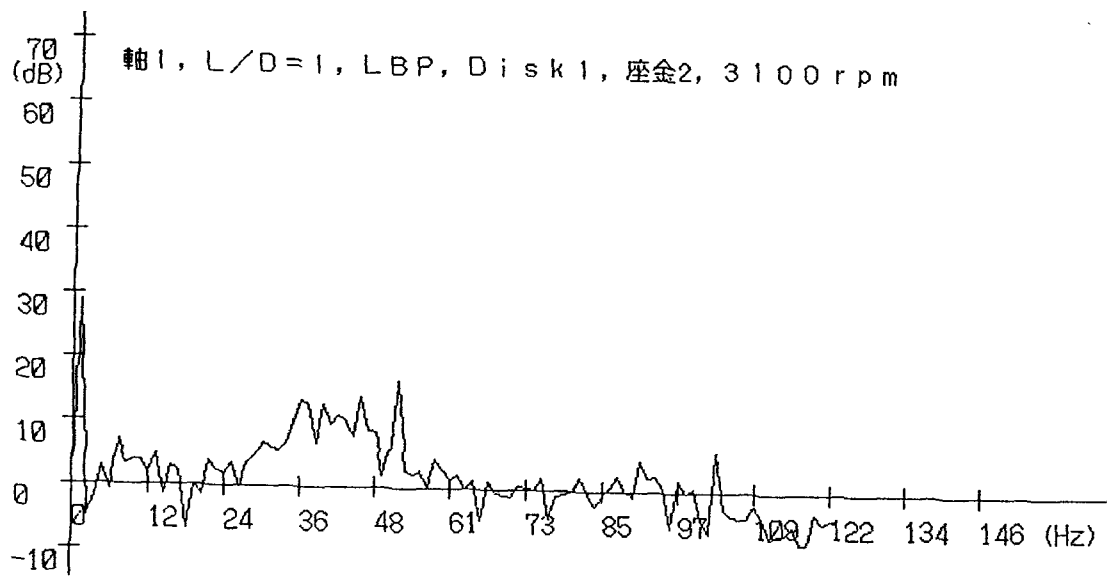
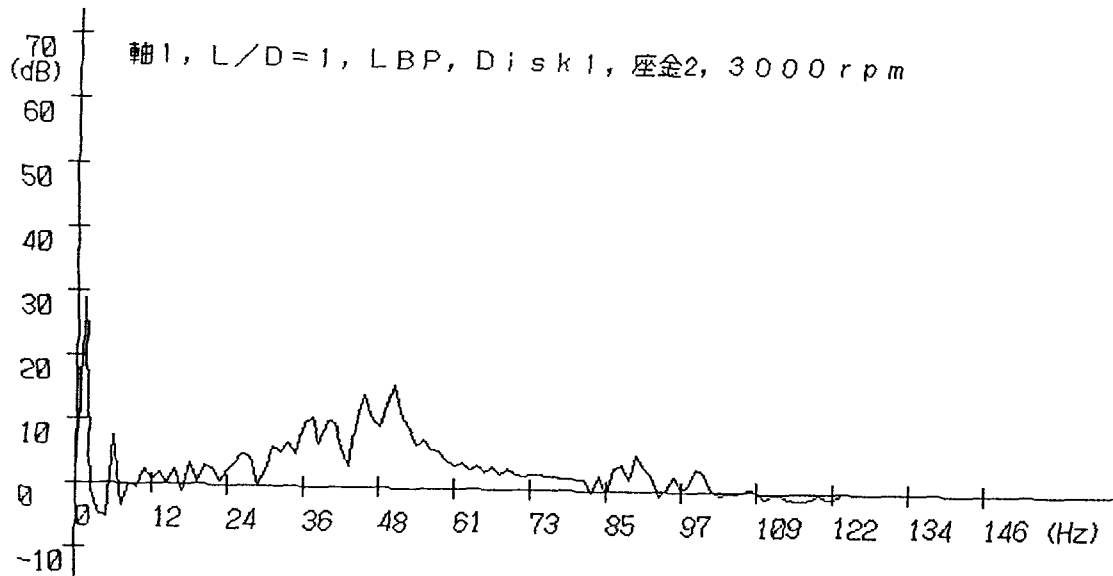
安定限界速度近辺の軸振動振幅



安定限界速度近辺の軸振動振幅

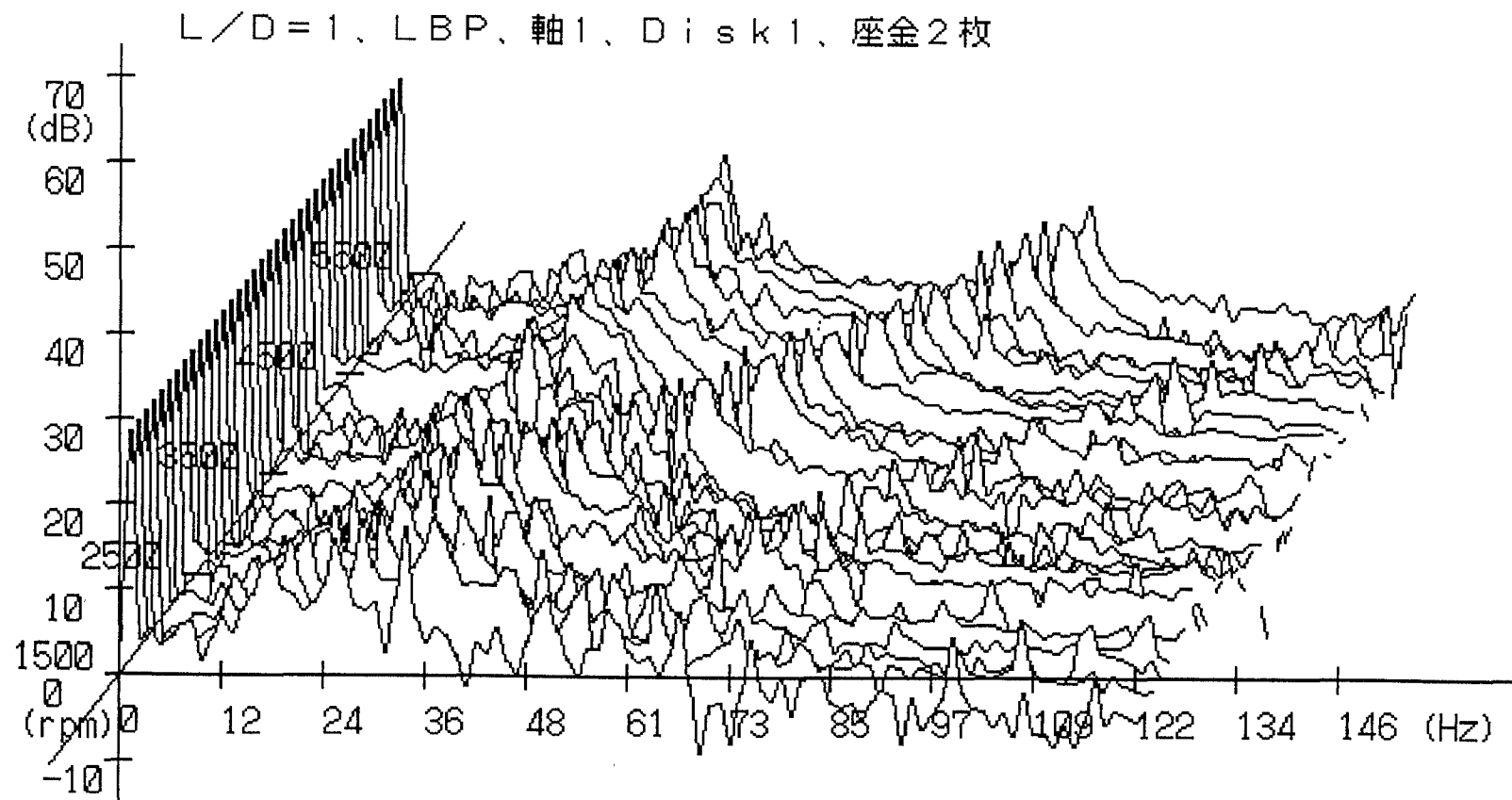


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

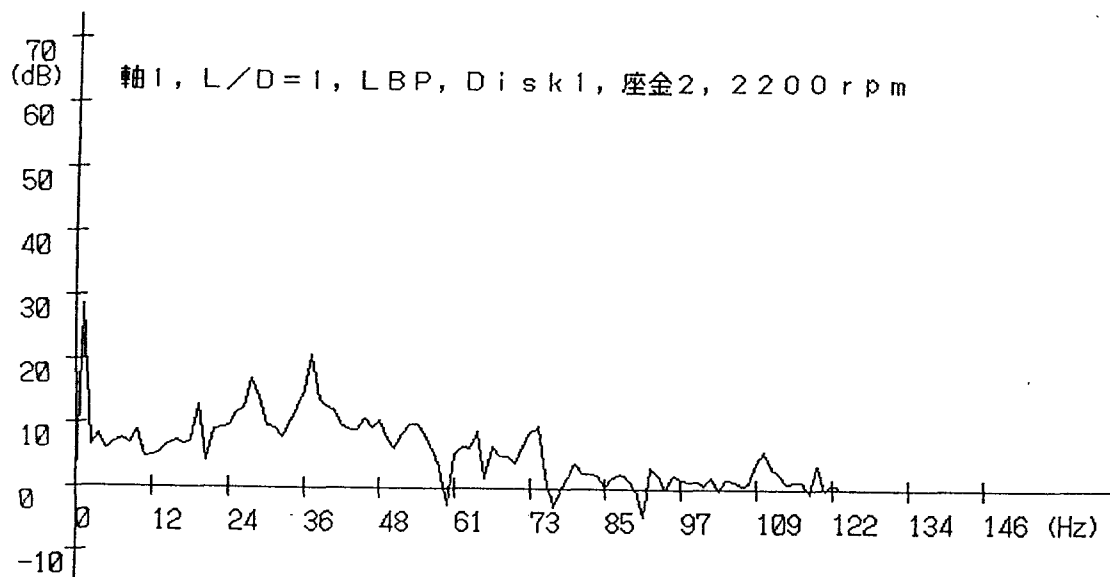
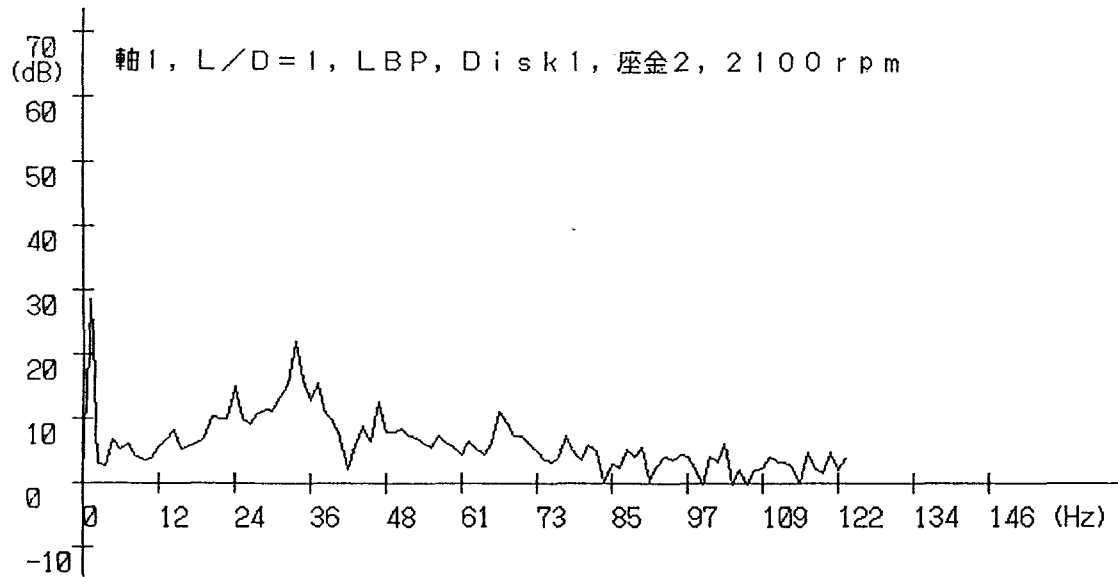


安定限界速度近辺の軸振動振幅

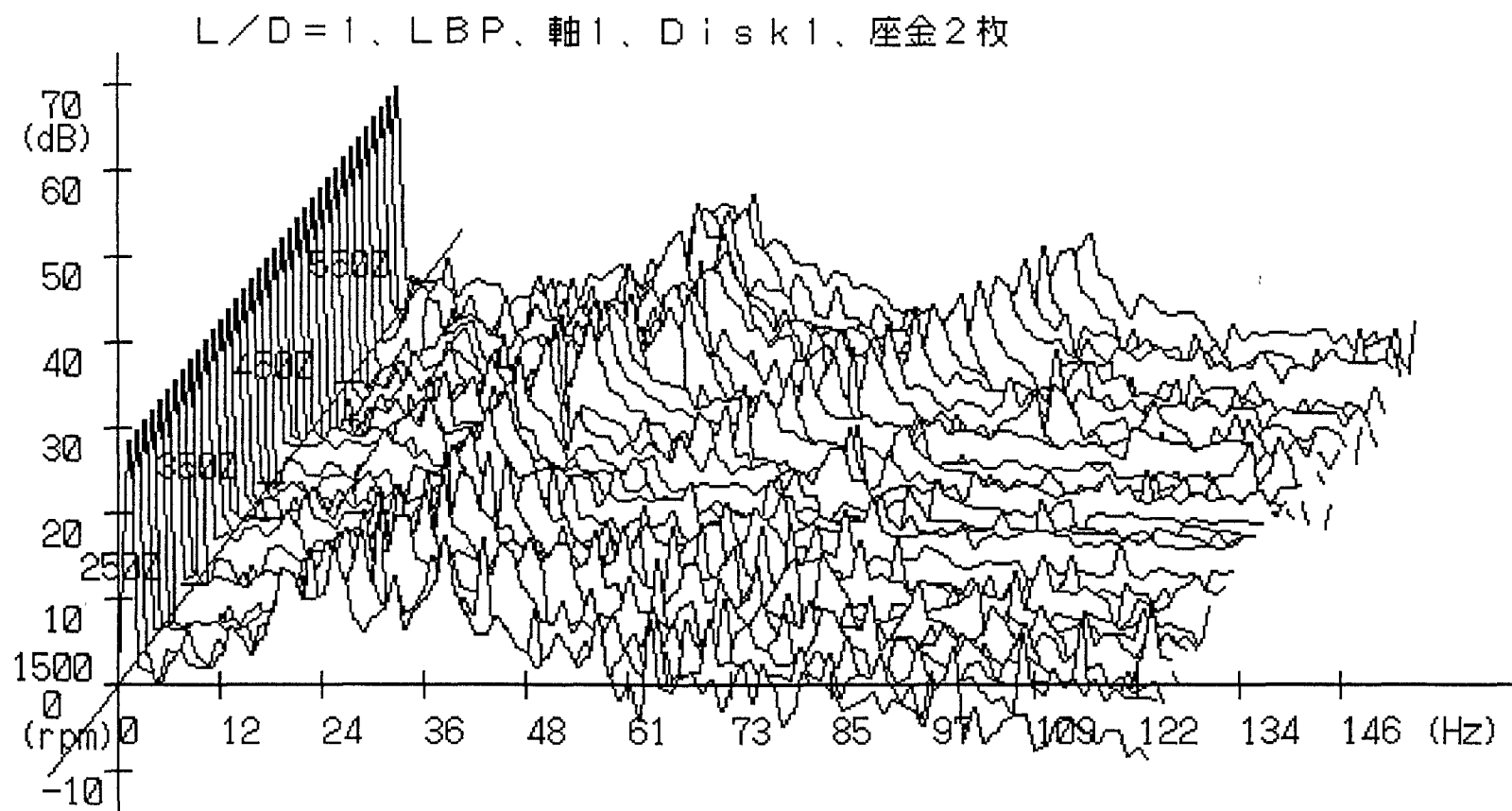




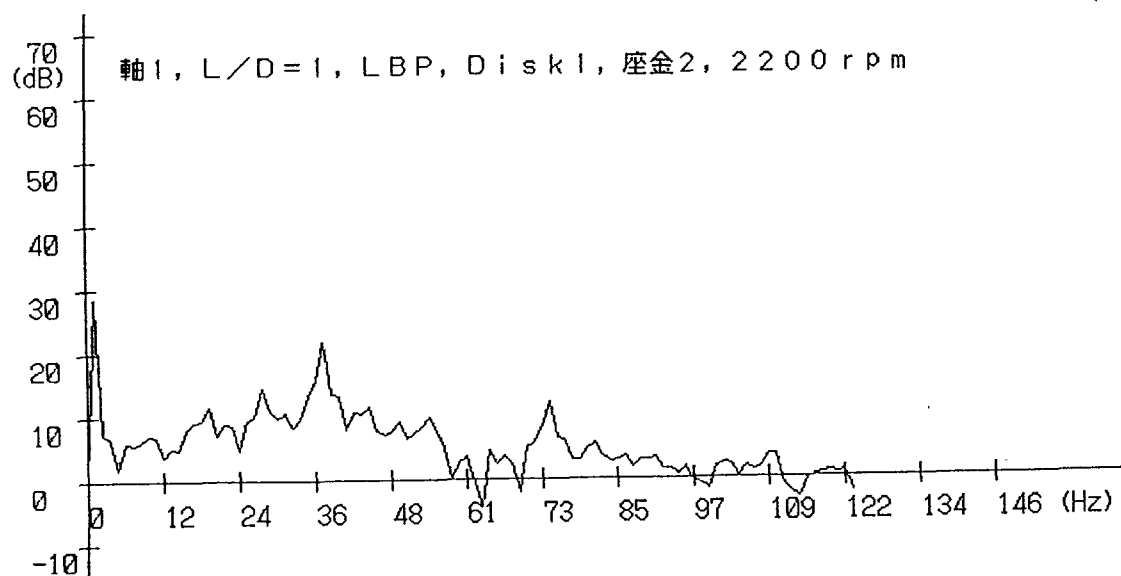
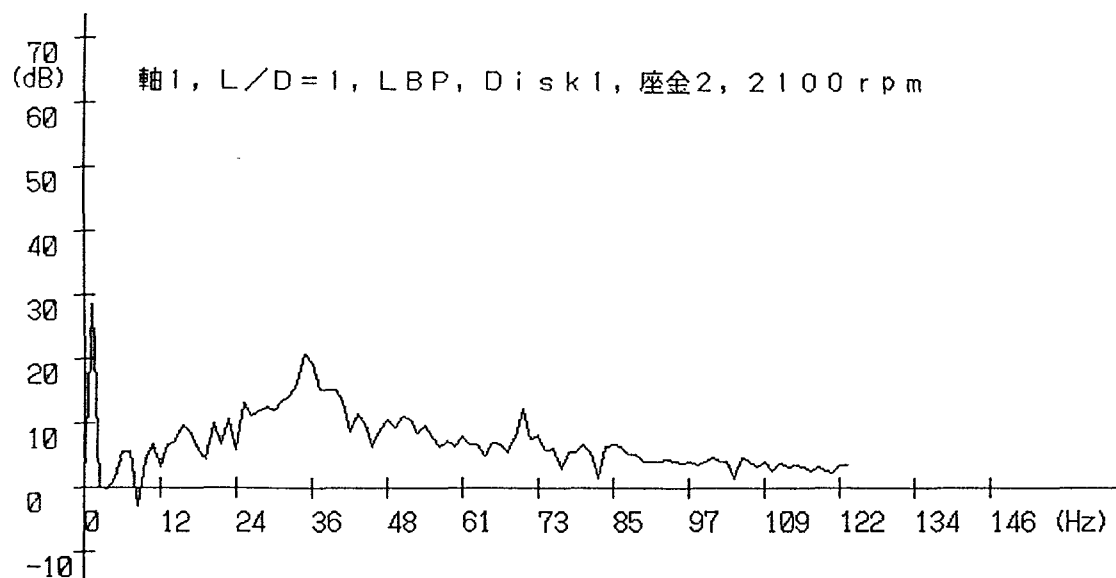
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



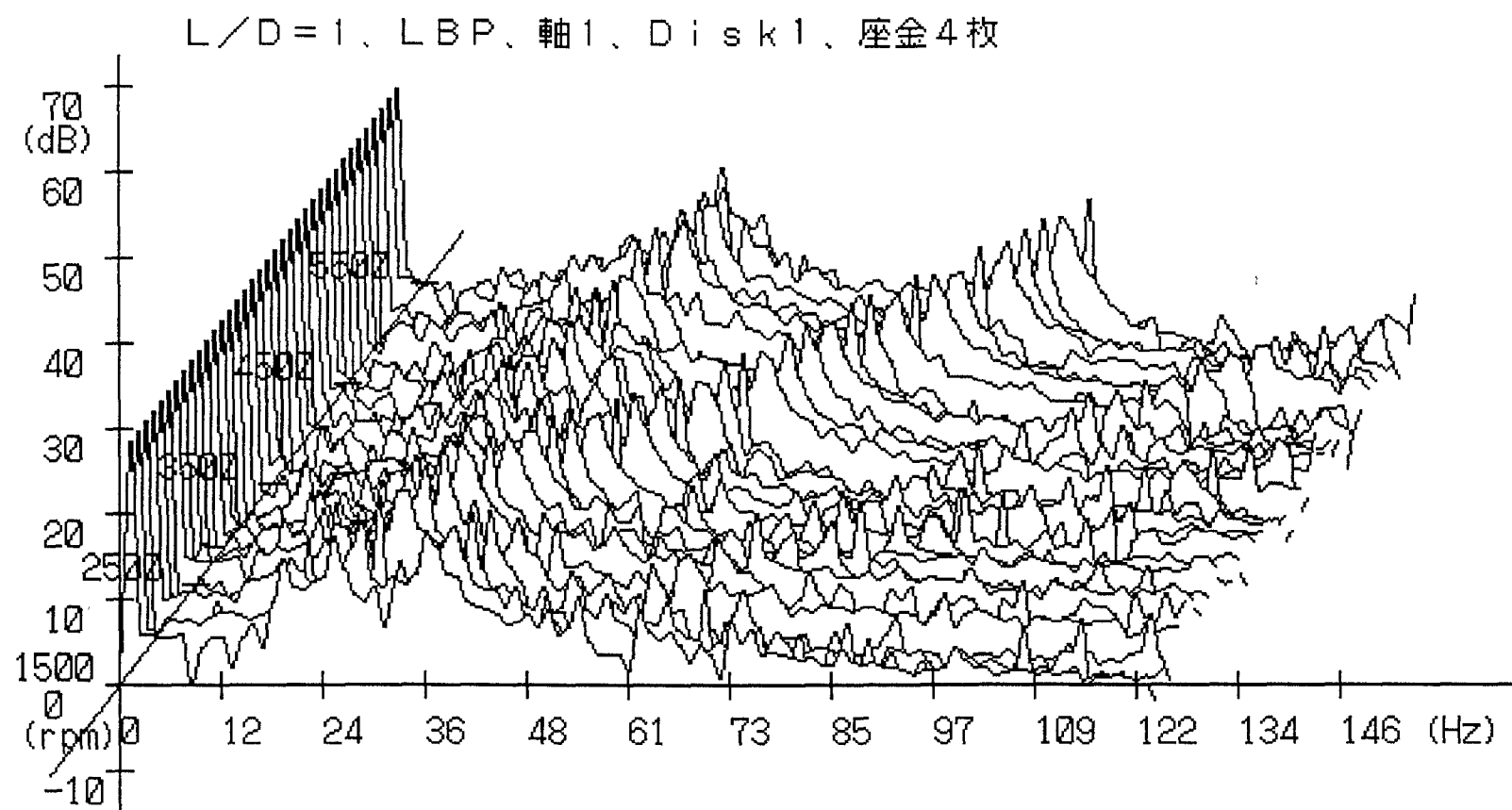
安定限界速度近辺の軸振動振幅



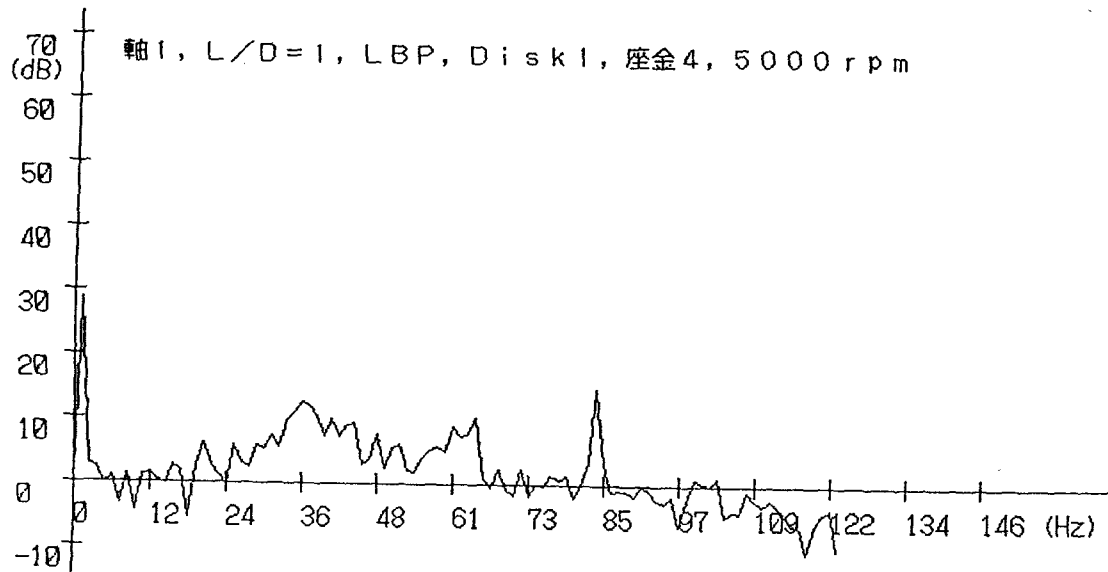
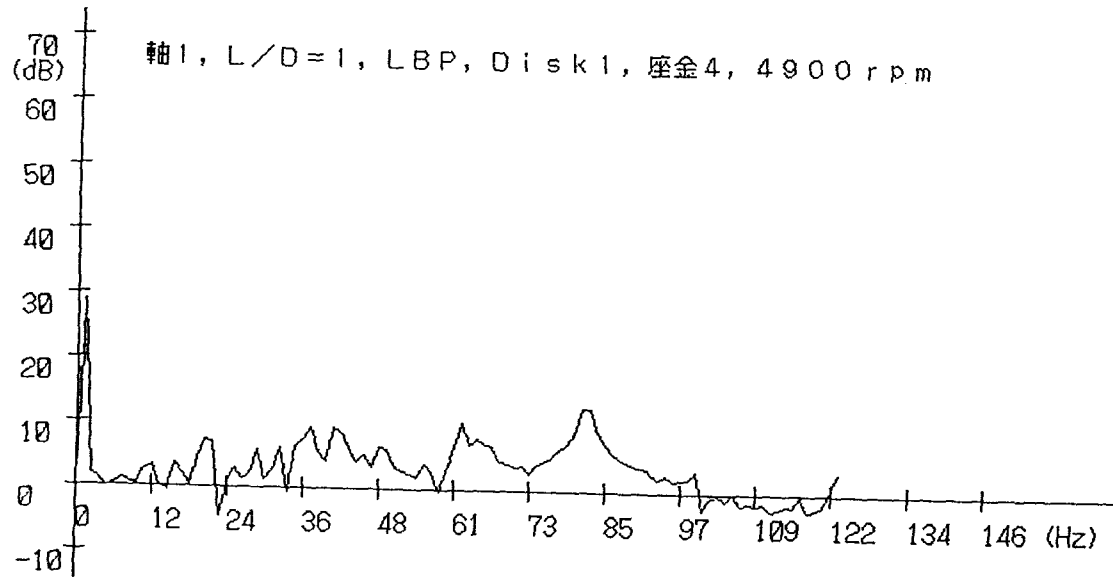
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



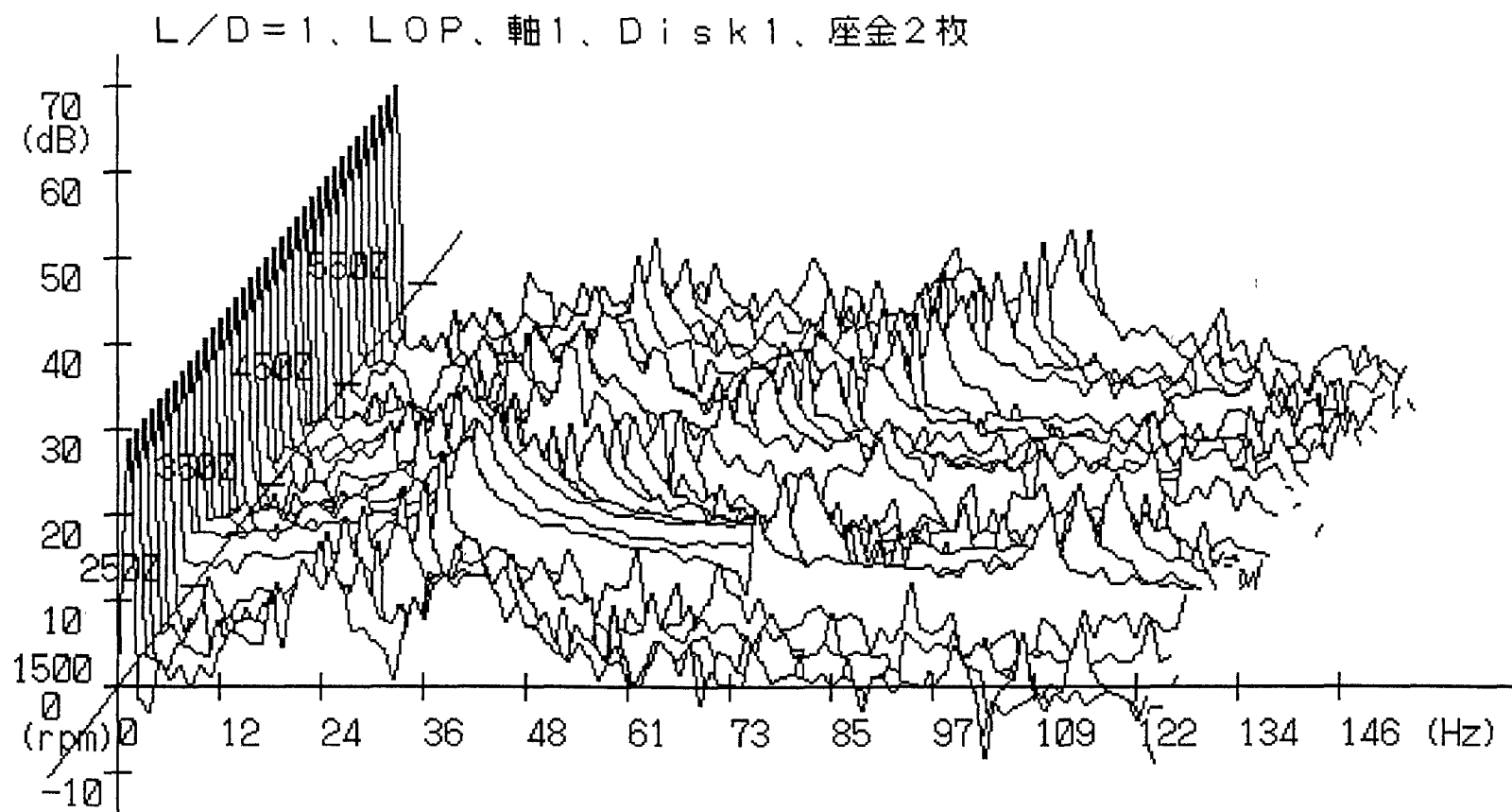
安定限界速度近辺の軸振動振幅



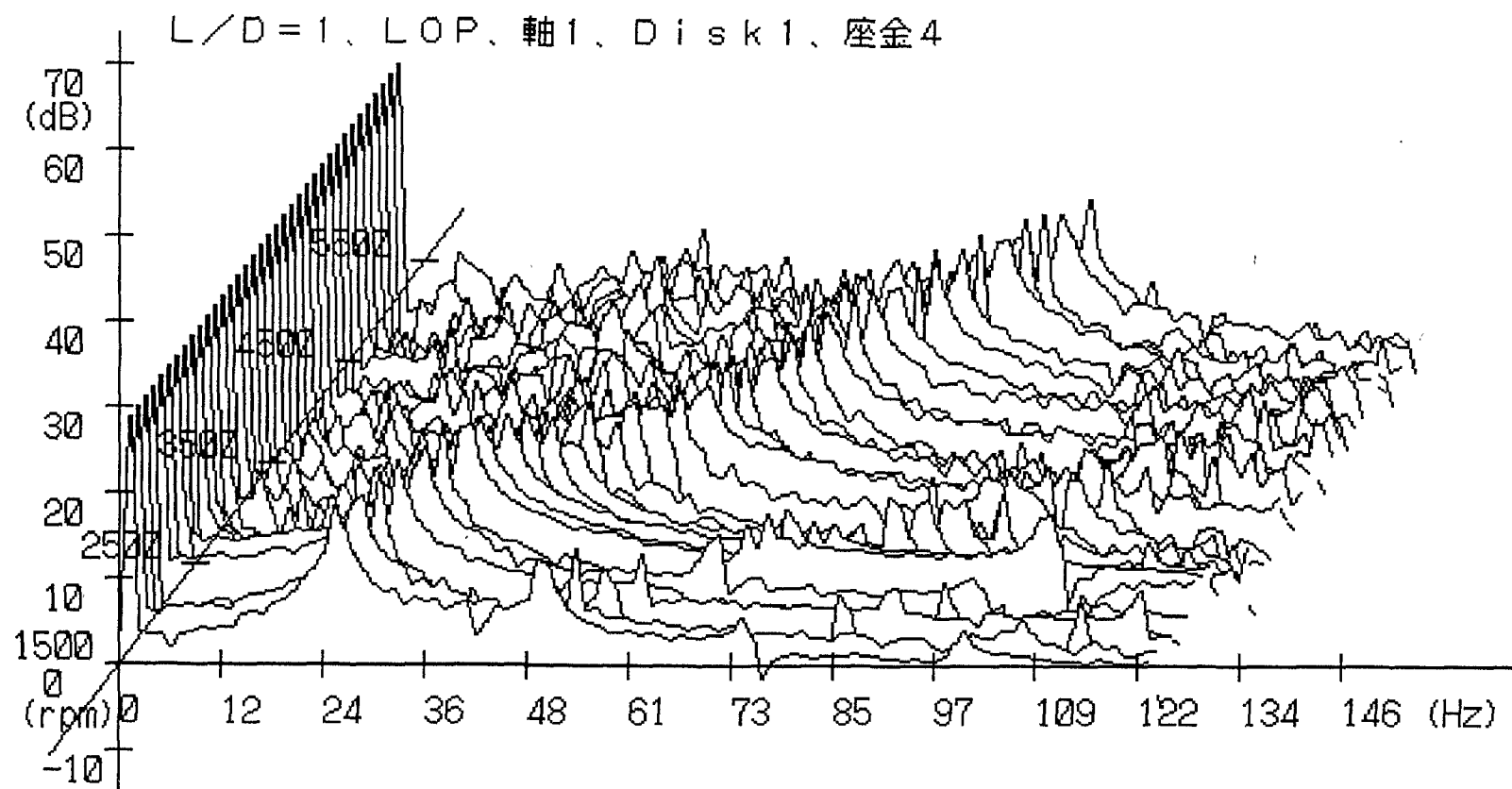
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



安定限界速度近辺の軸振動振幅

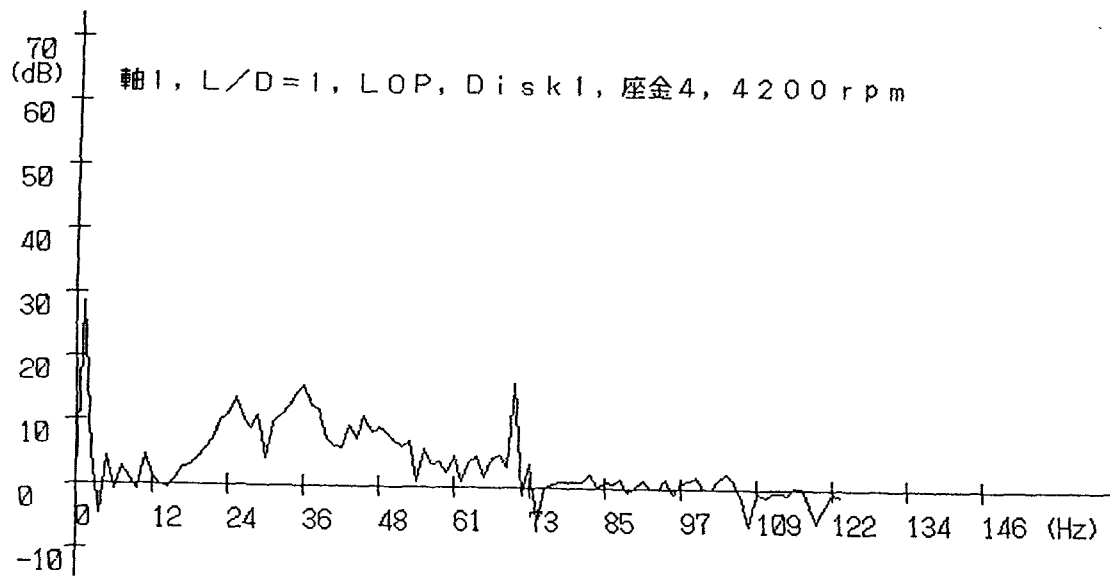
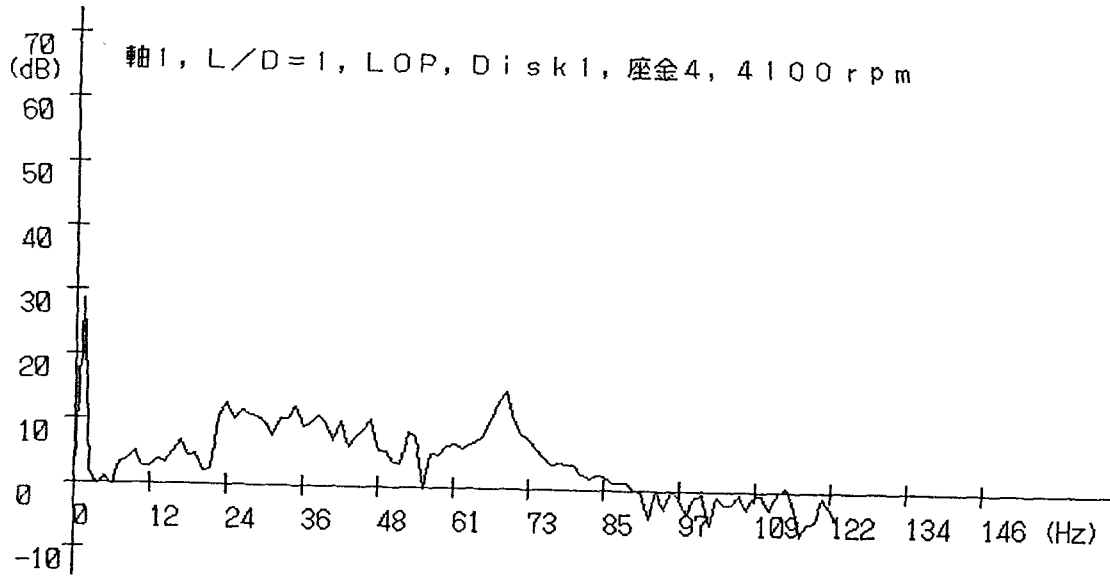


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

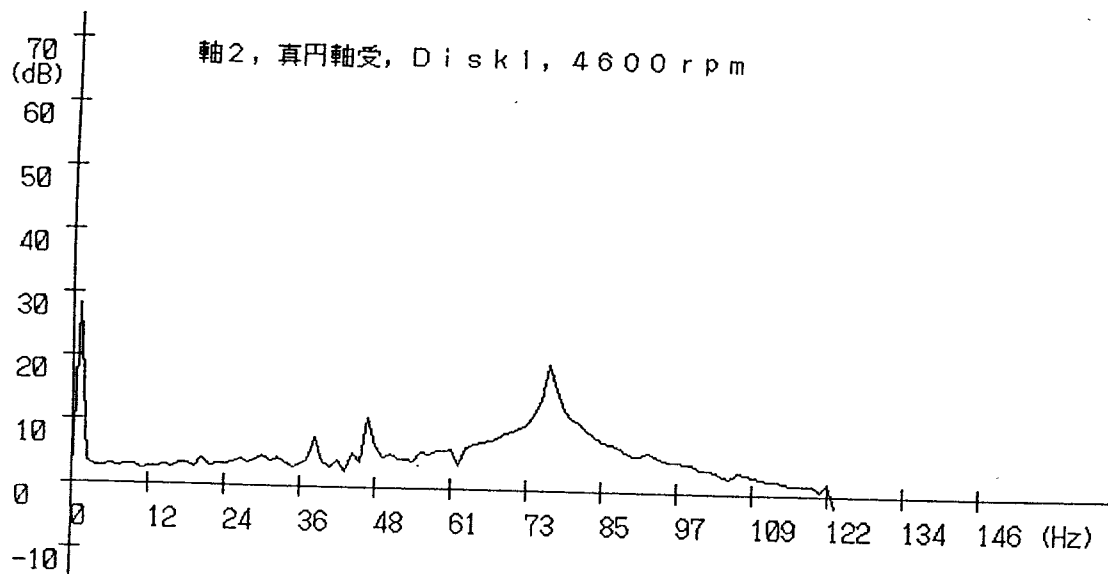
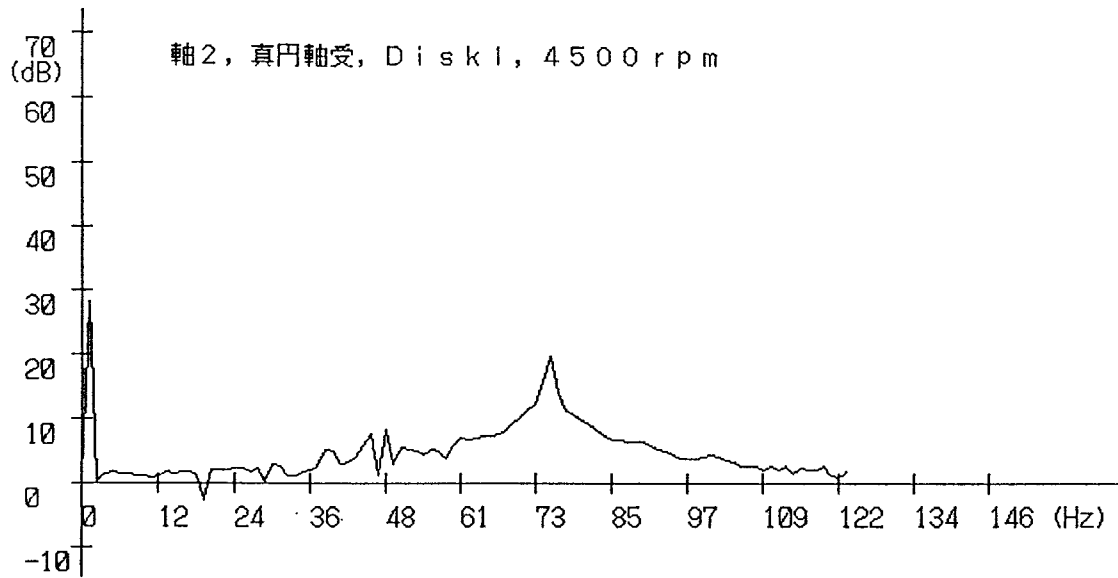


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

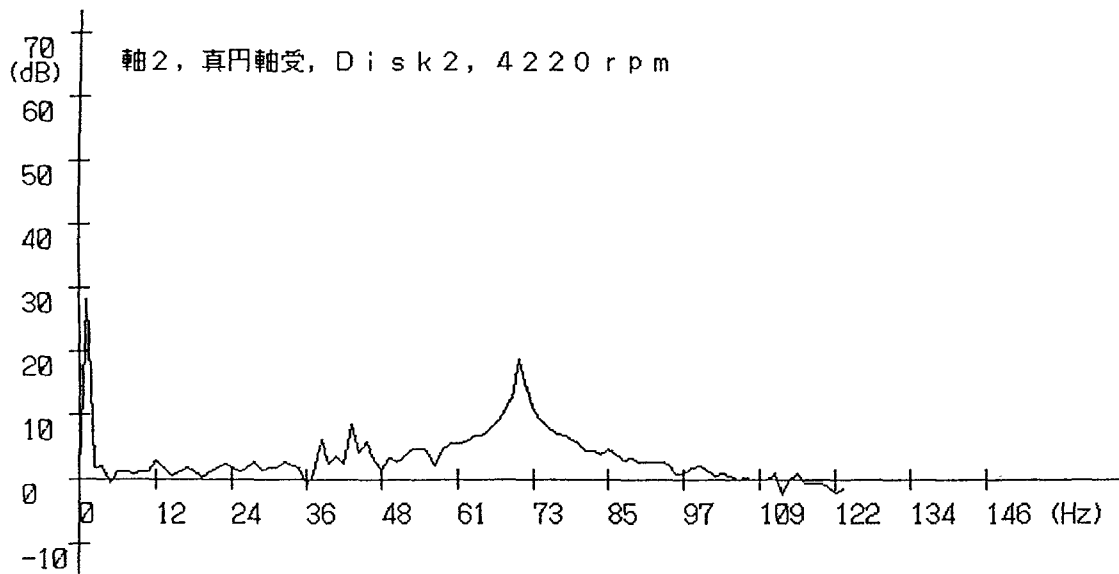
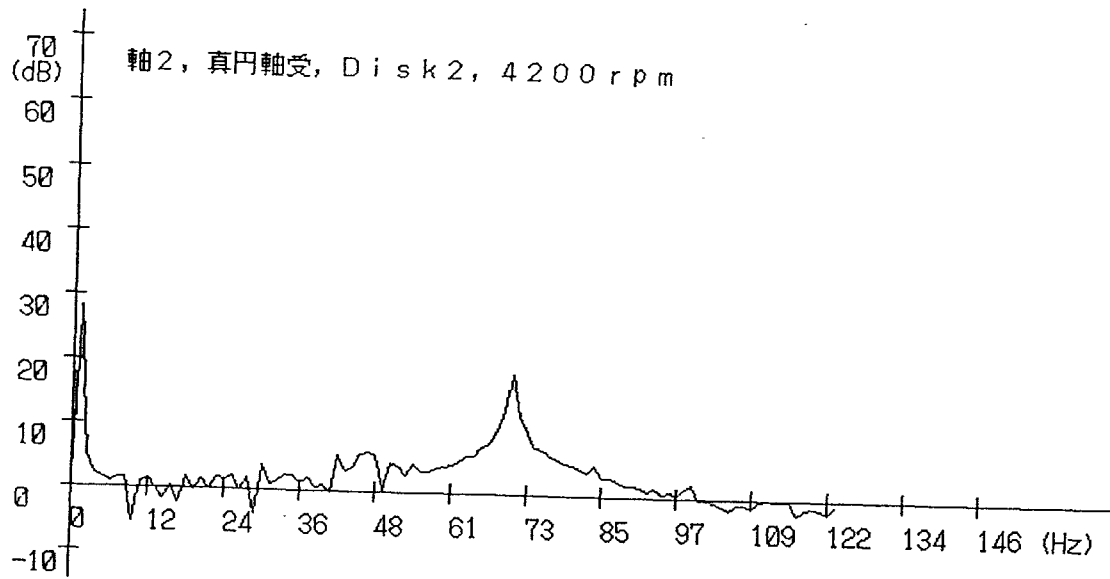




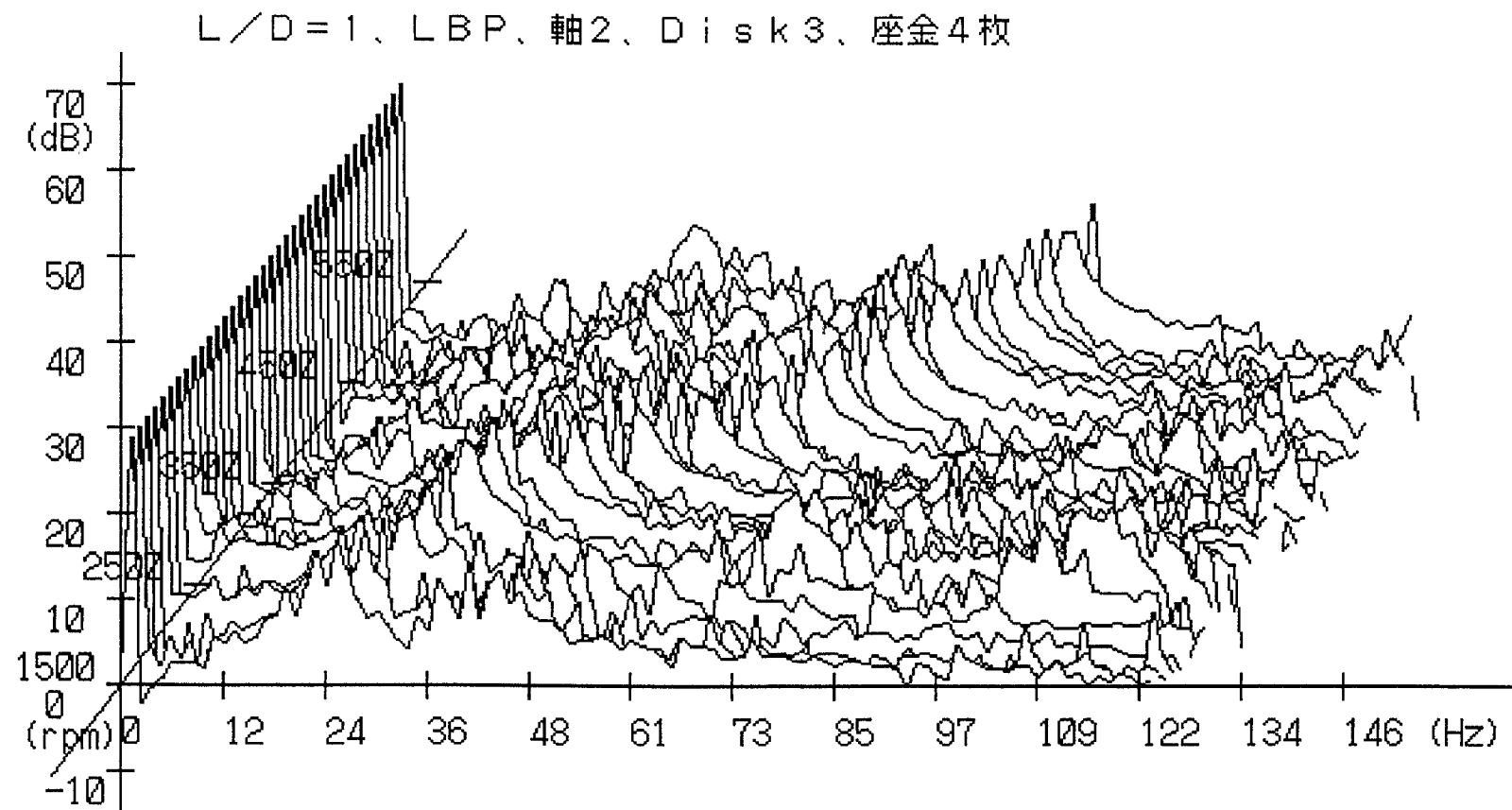
安定限界速度近辺の軸振動振幅



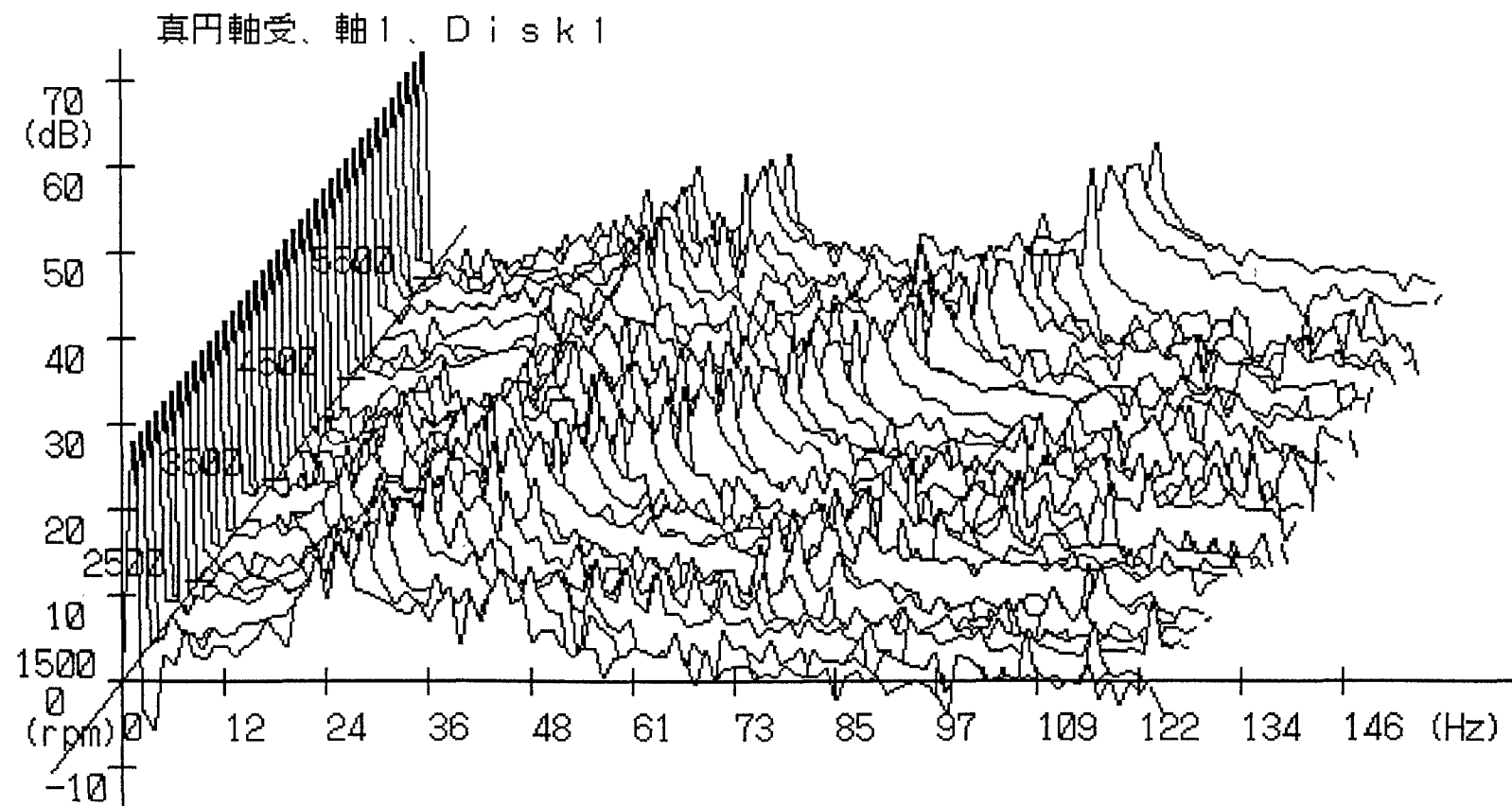
安定限界速度近辺の軸振動振幅



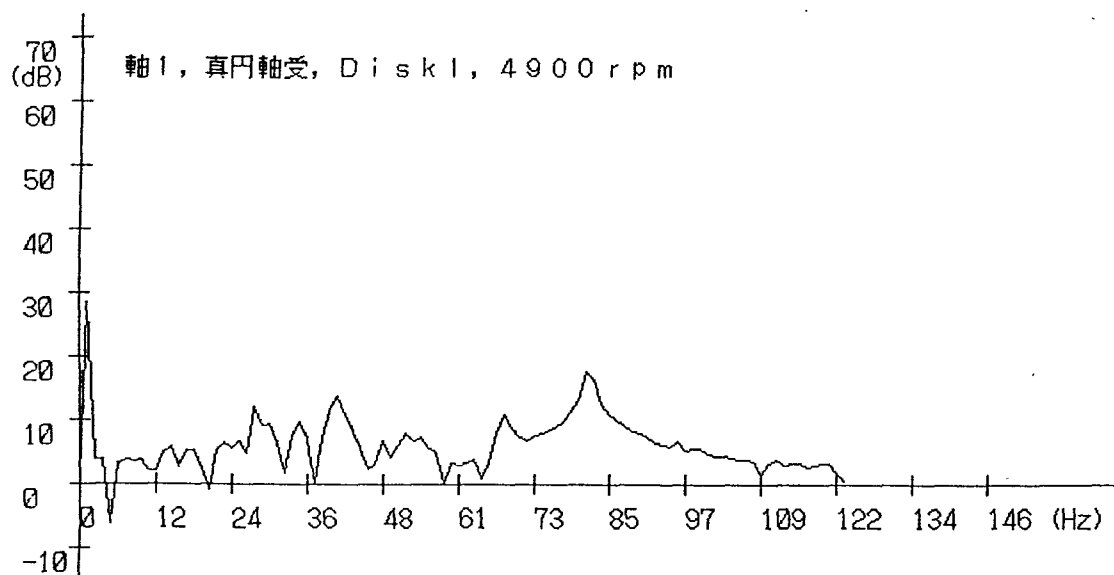
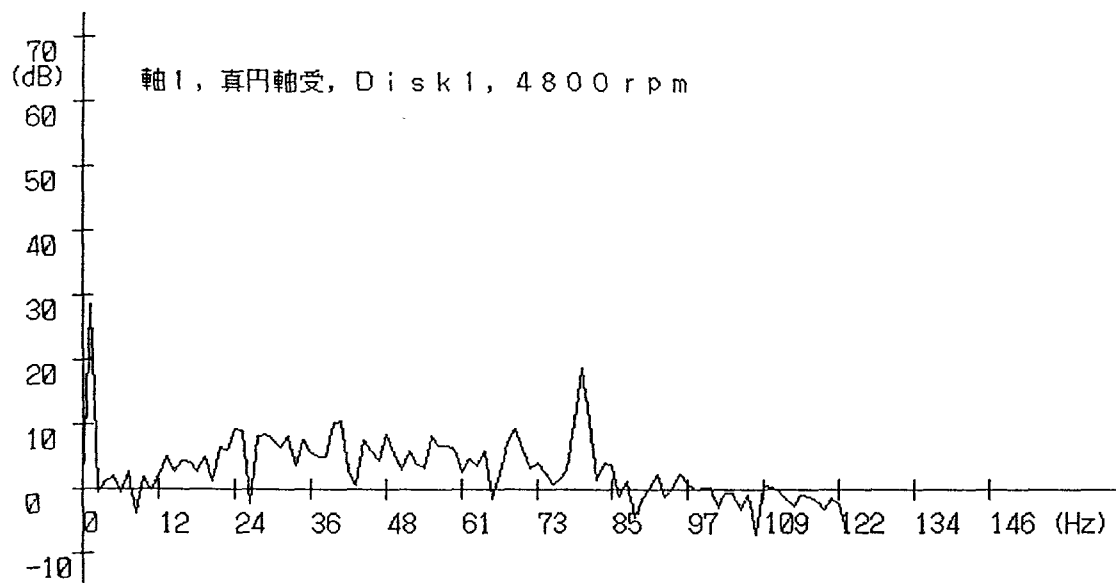
安定限界速度近辺の軸振動振幅



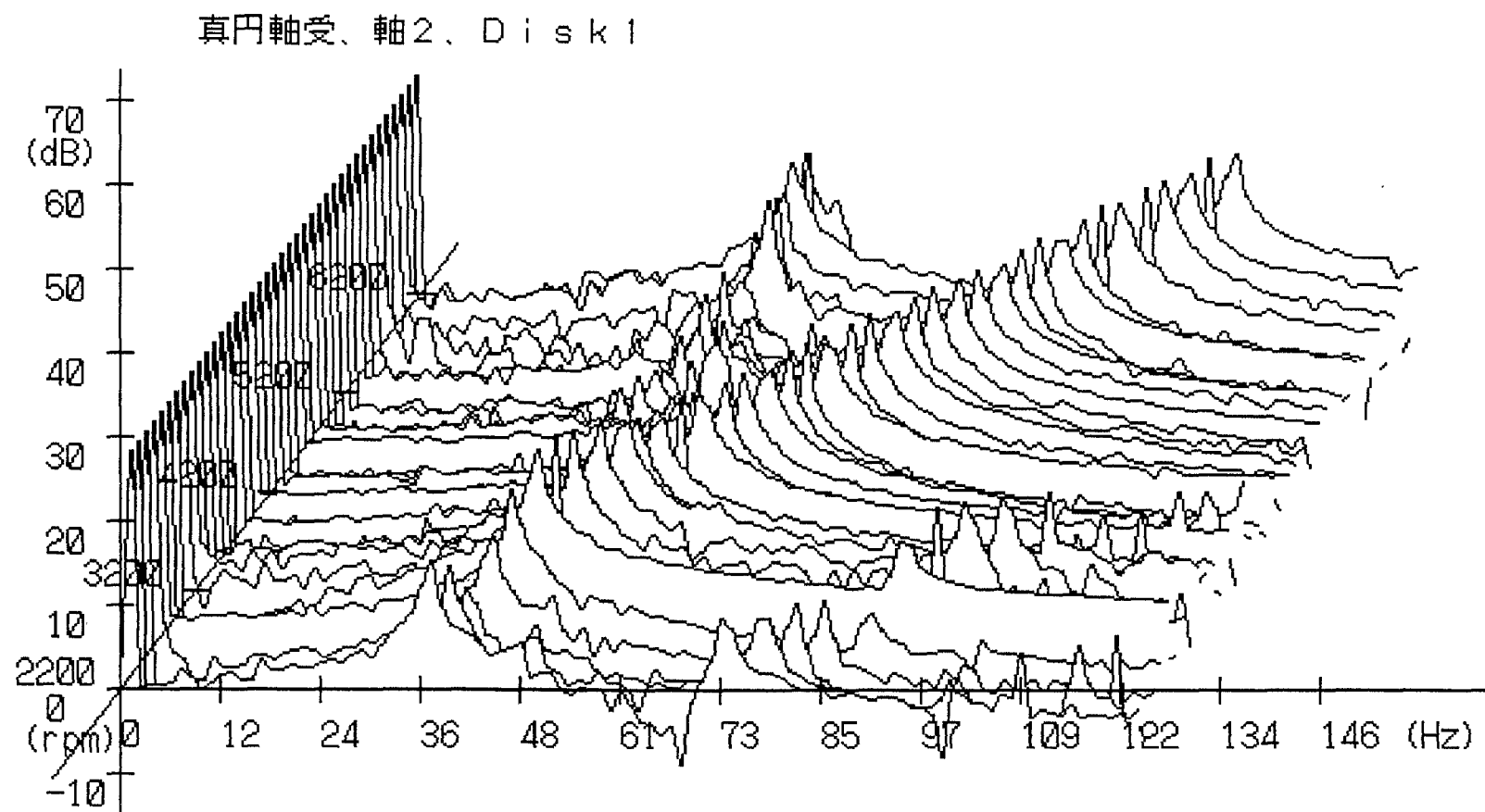
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



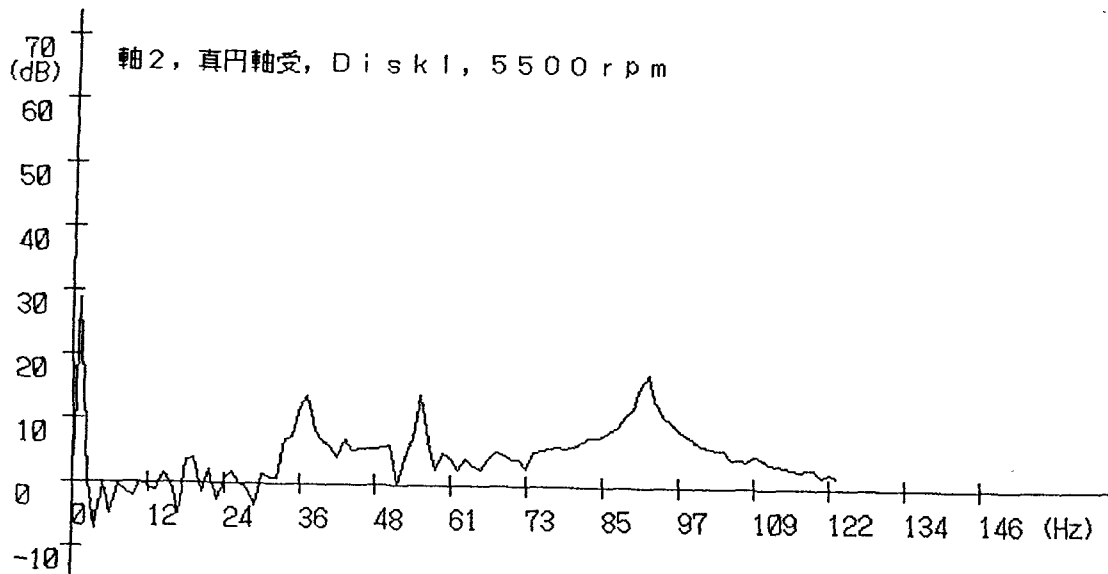
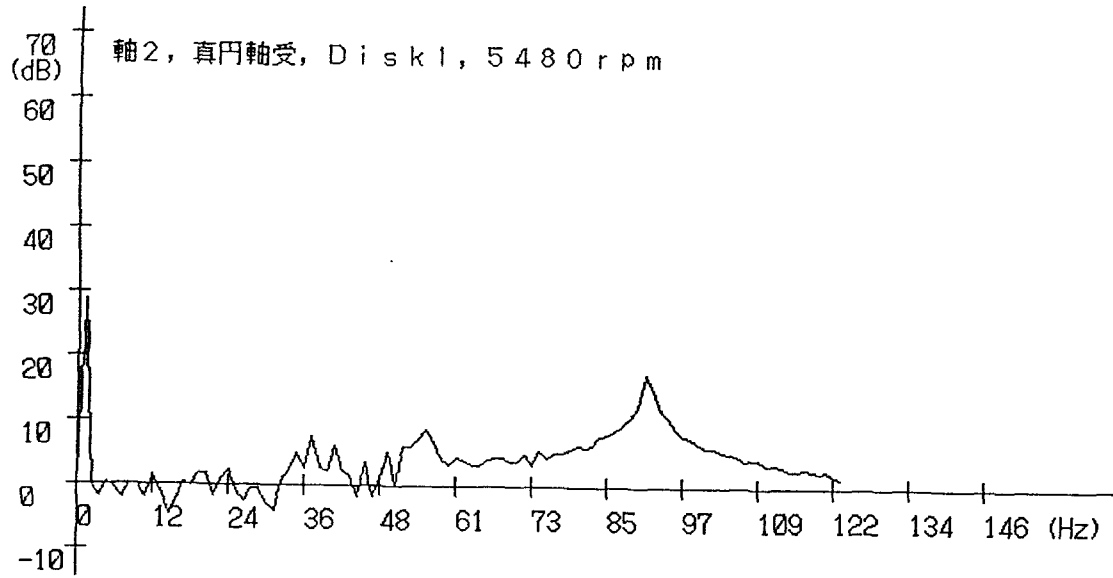
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



安定限界速度近辺の軸振動振幅

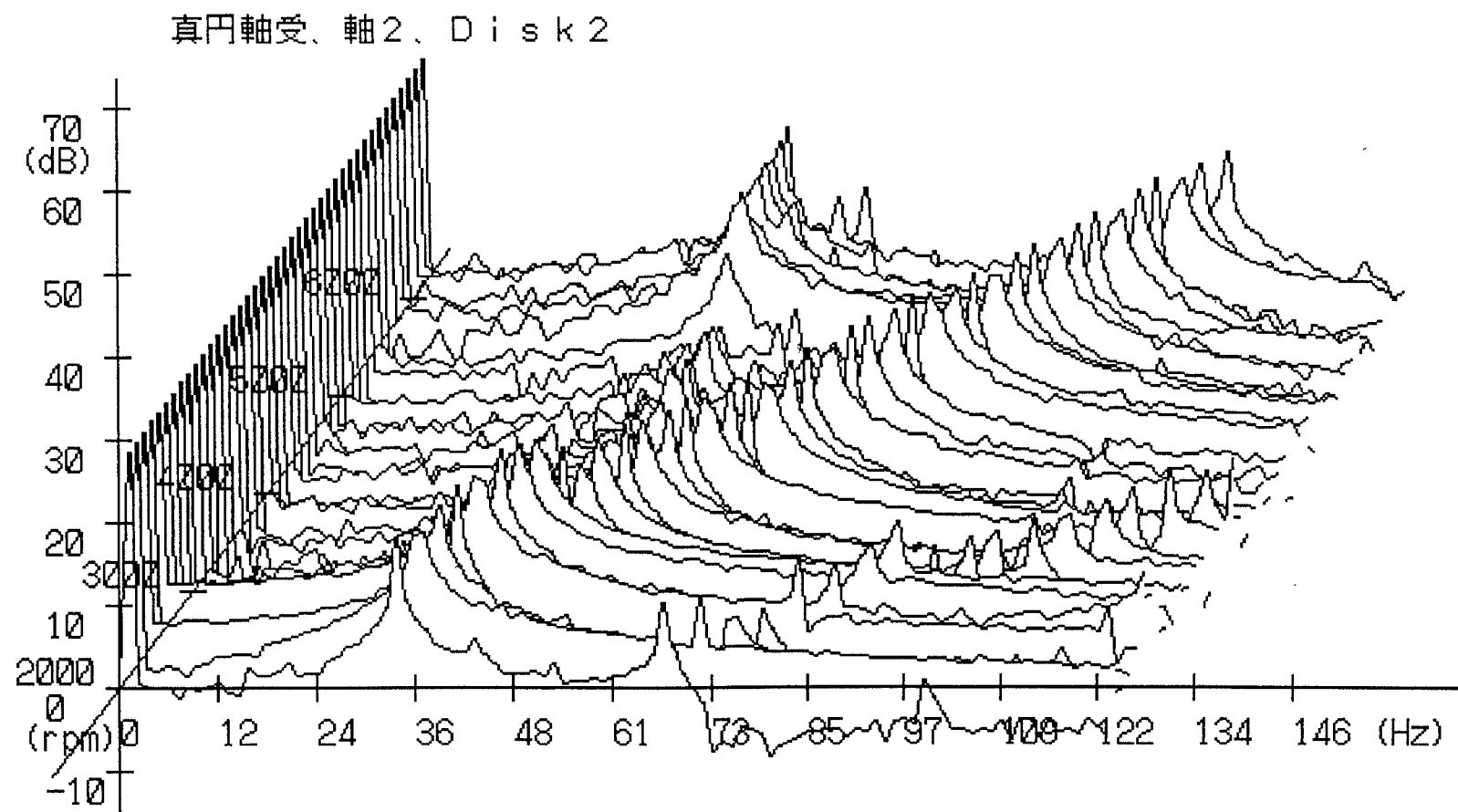


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

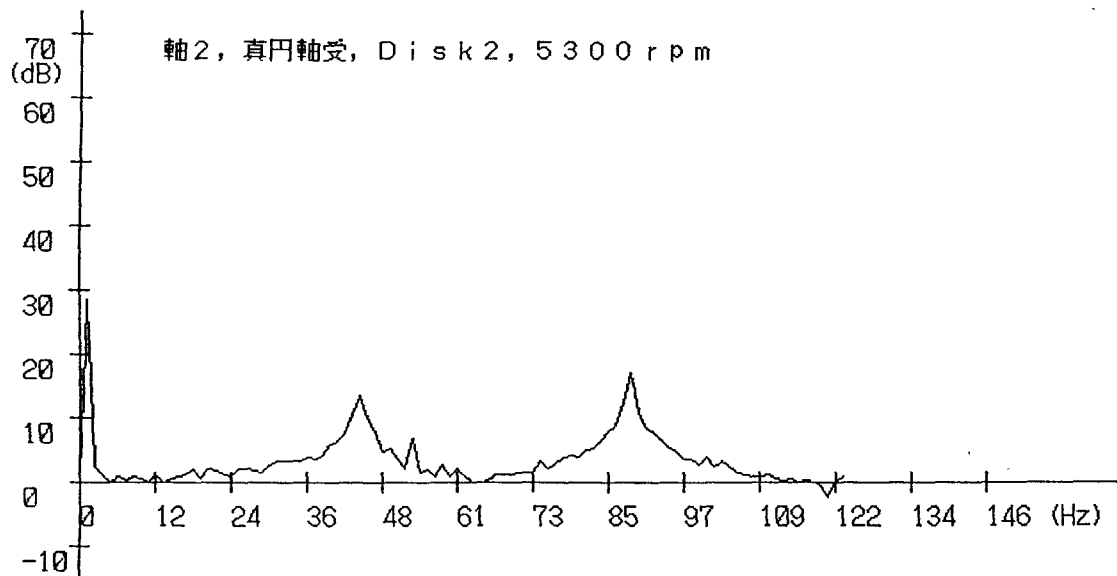
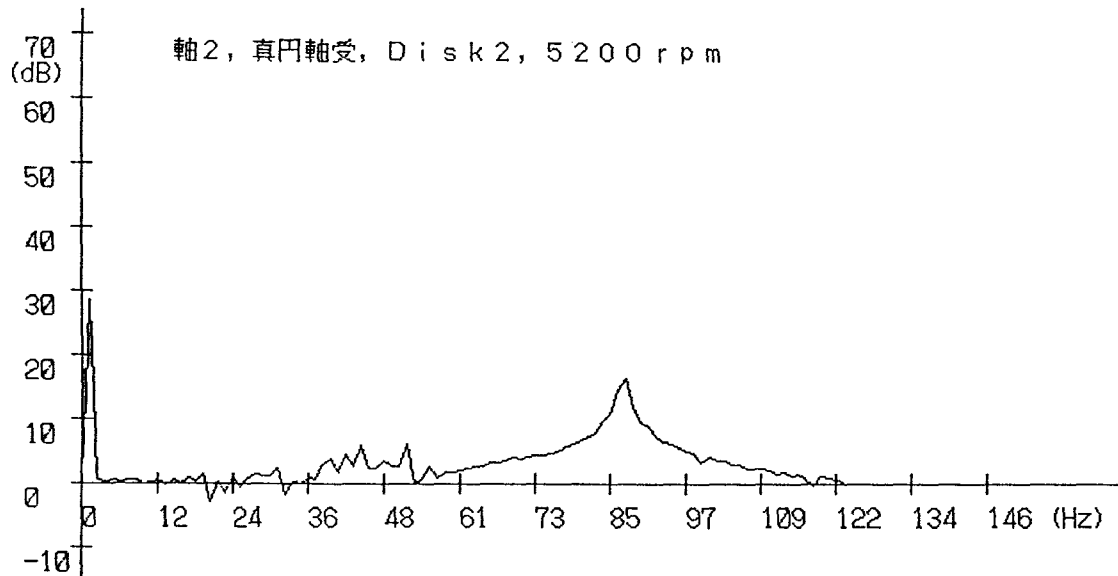


安定限界速度近辺の軸振動振幅

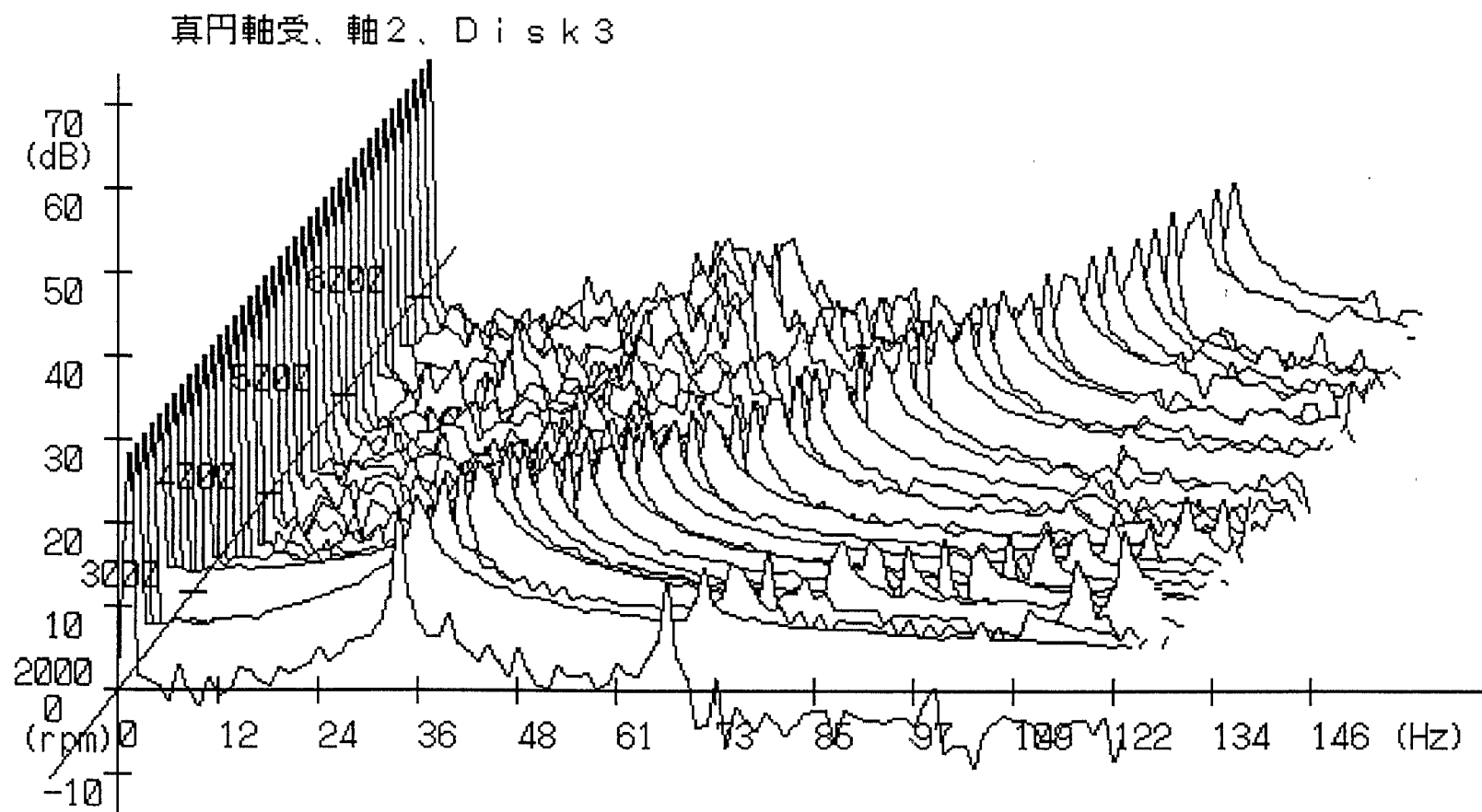




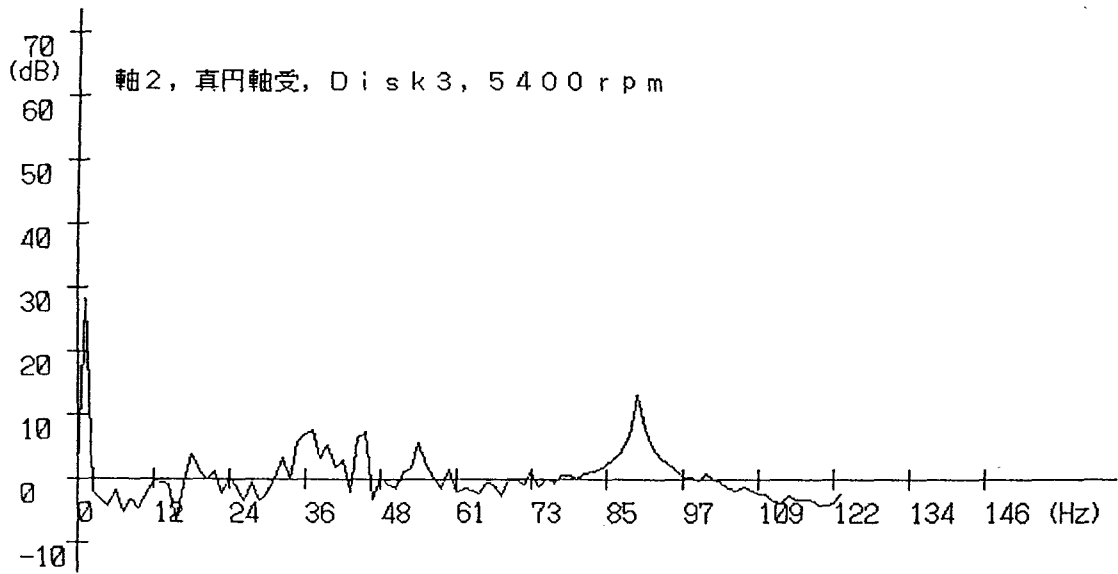
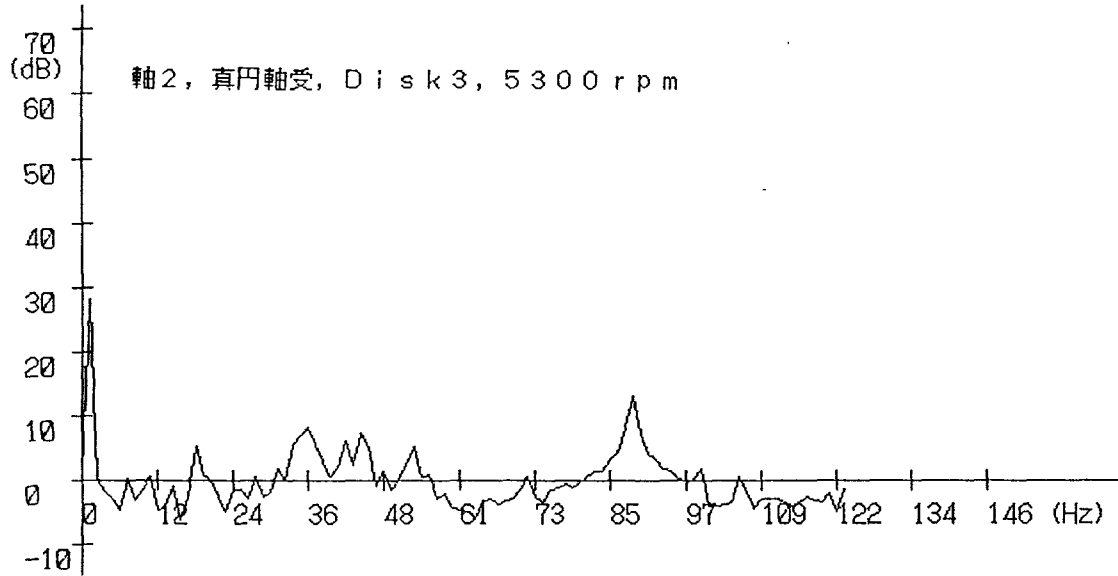
回転速度、周波数をパラメータとしたときの振動振幅の解析結果



安定限界速度近辺の軸振動振幅

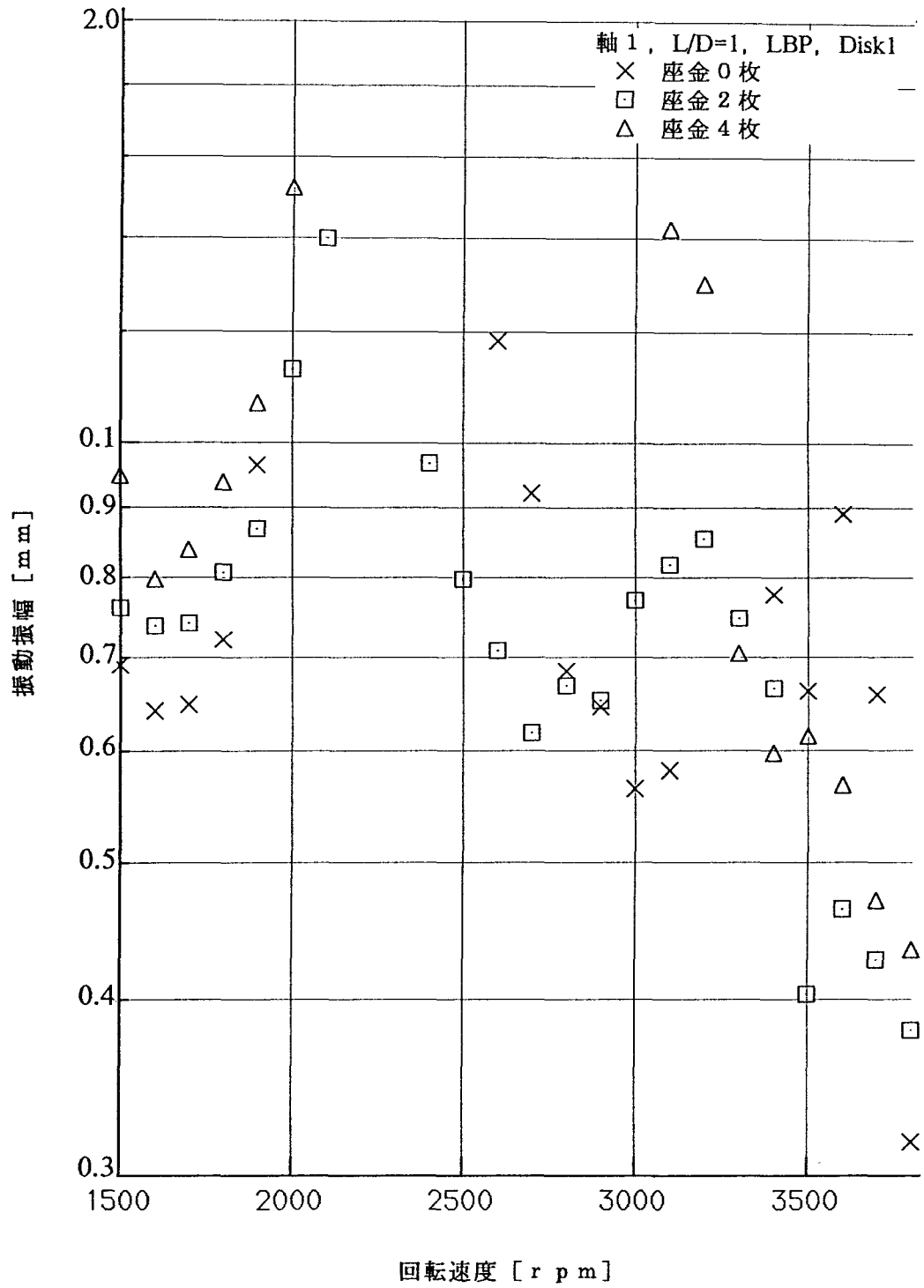


回転速度、周波数をパラメータとしたときの振動振幅の解析結果

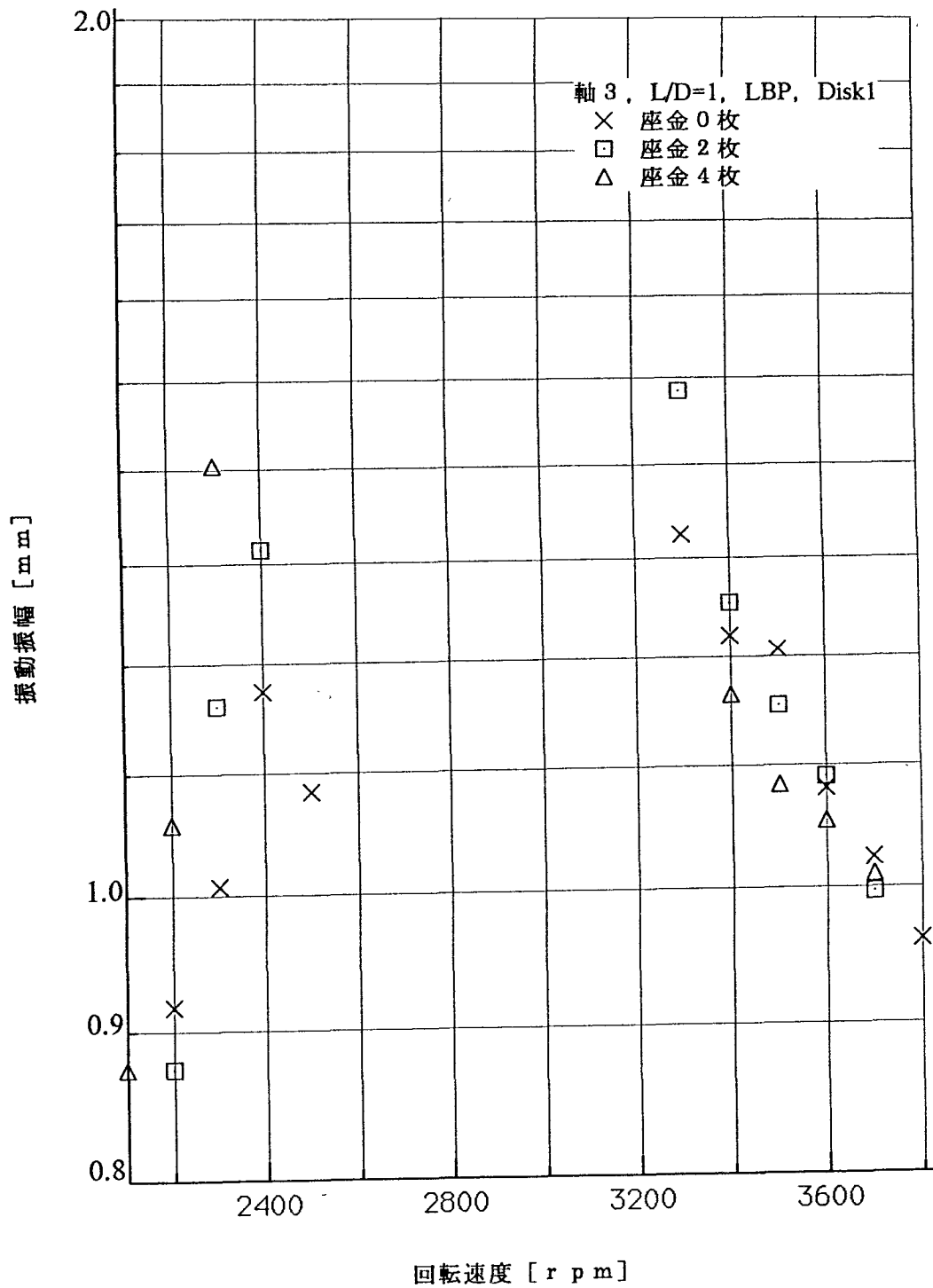


安定限界速度近辺の軸振動振幅

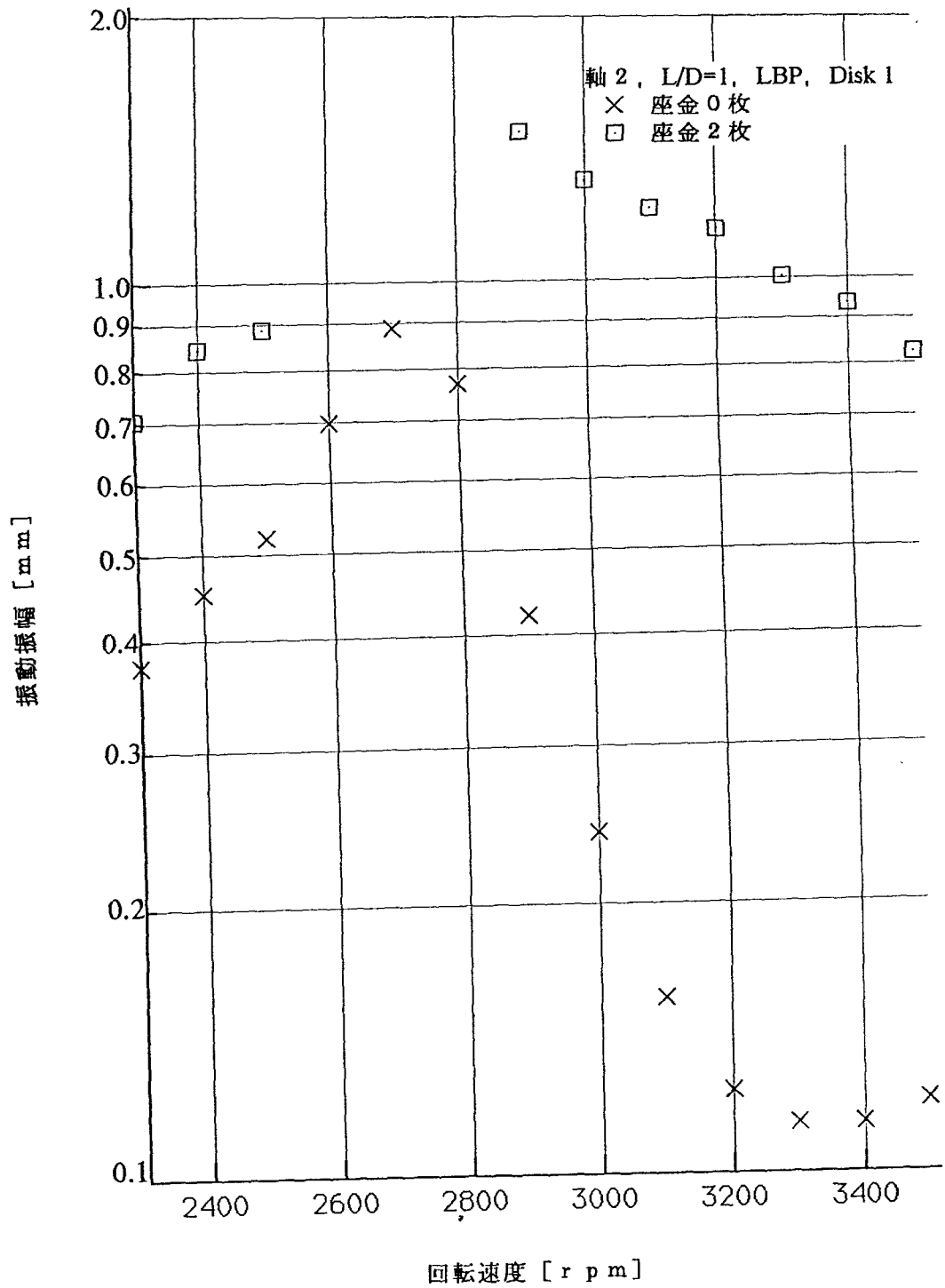
## 付録 4 実験結果（本文に載せた結果以外）



実験結果（本文に載せた結果以外）

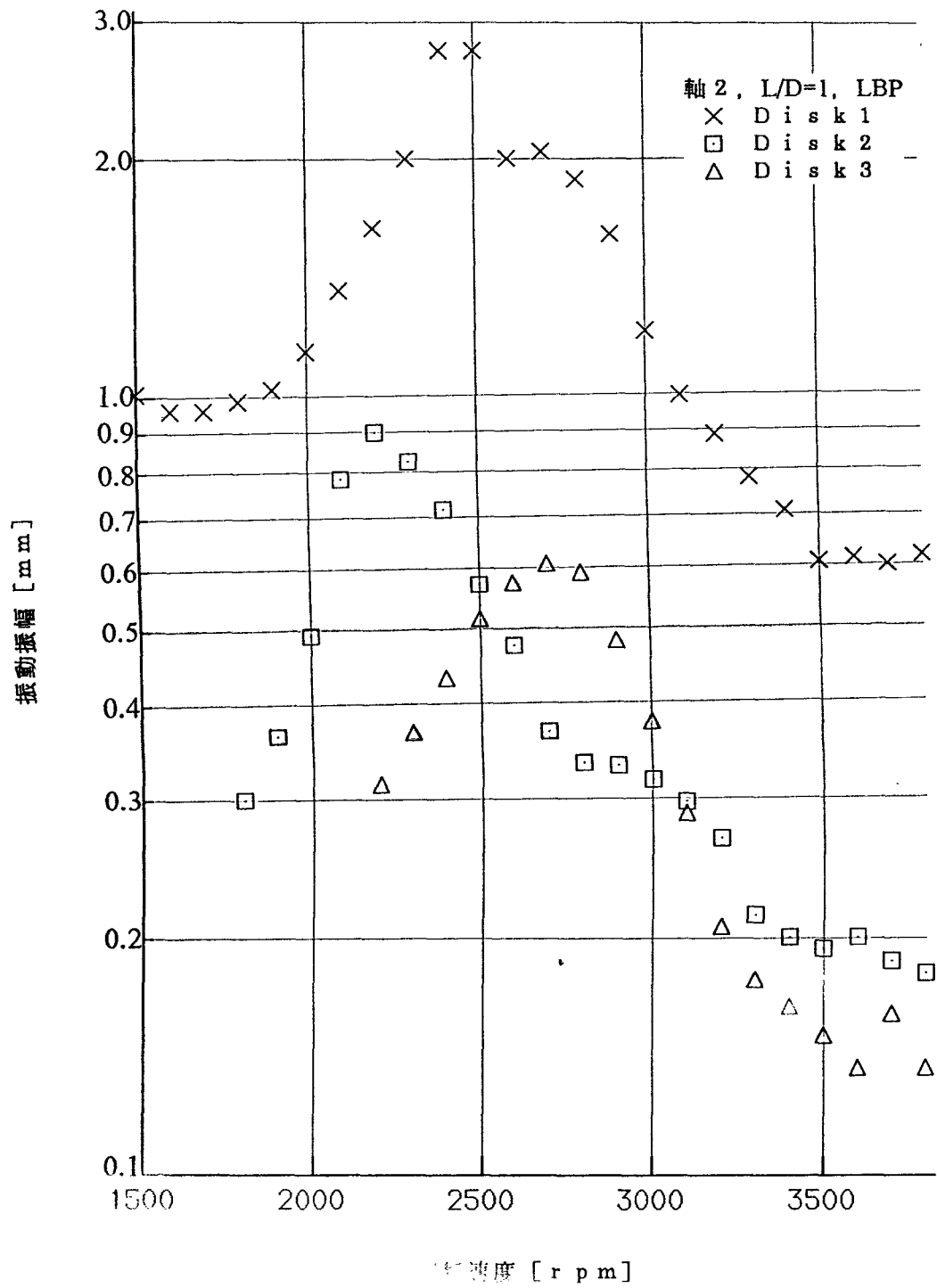


実験結果（本文に載せた結果以外）

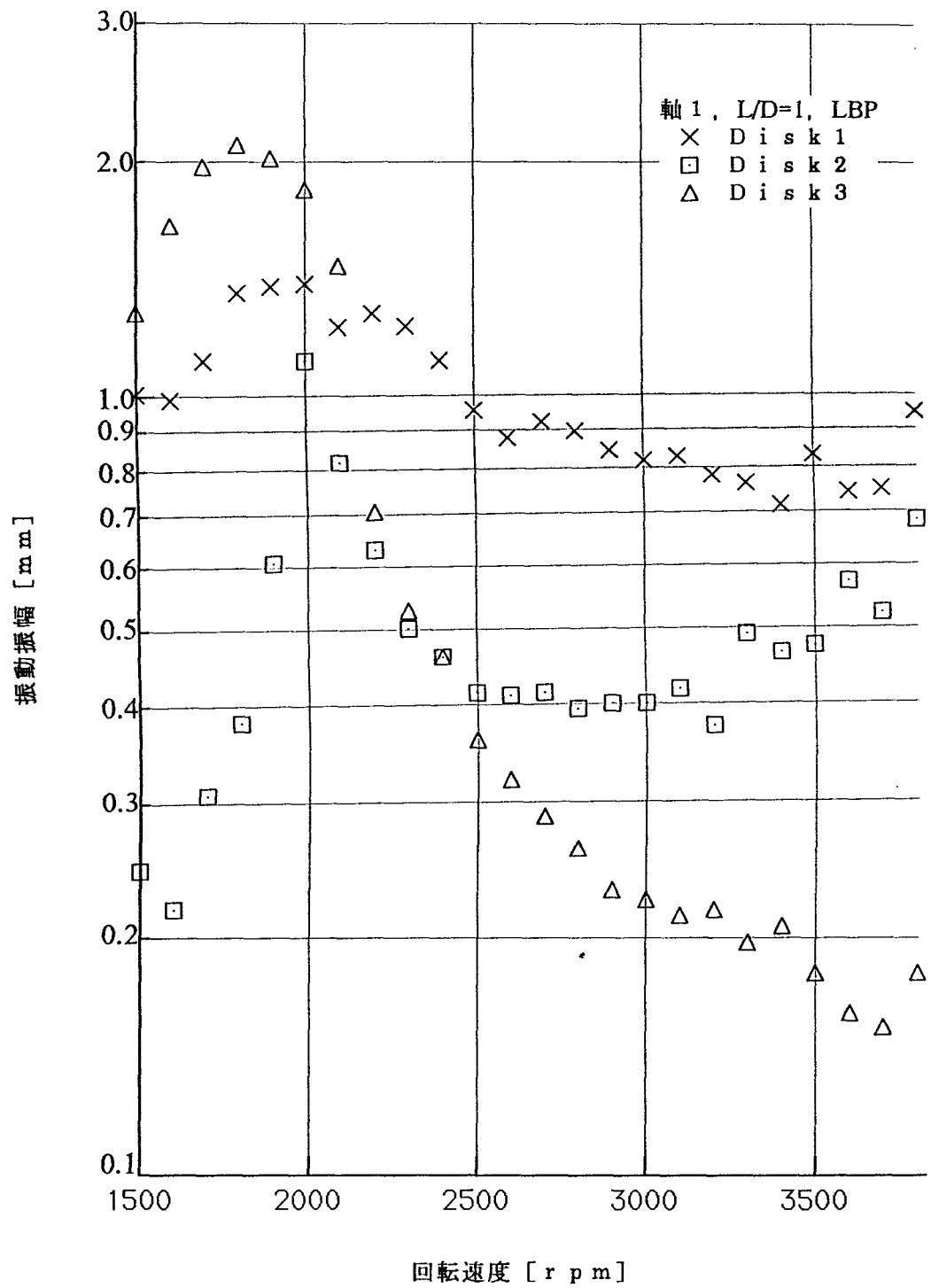


実験結果（本文に載せた結果以外）

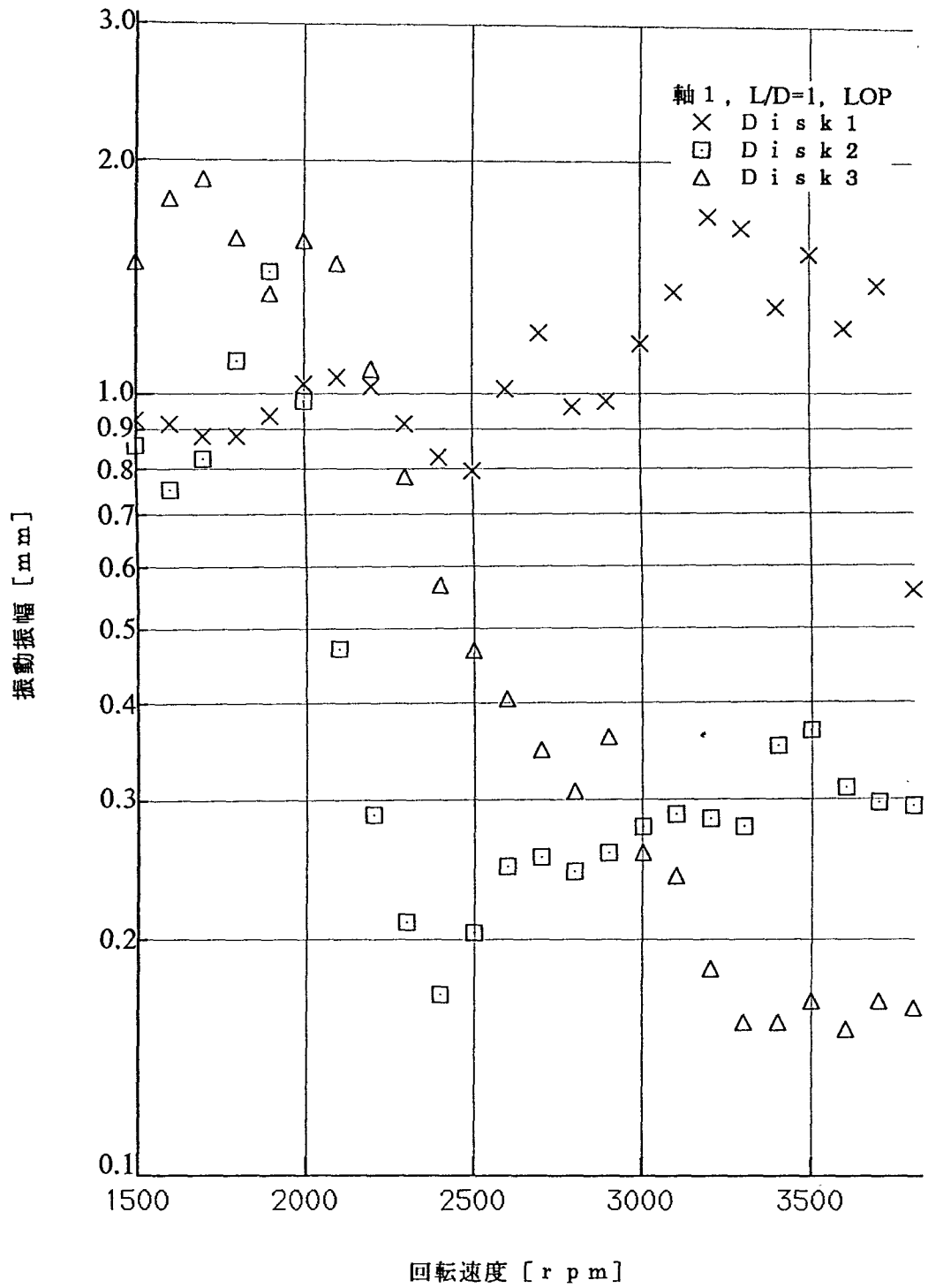




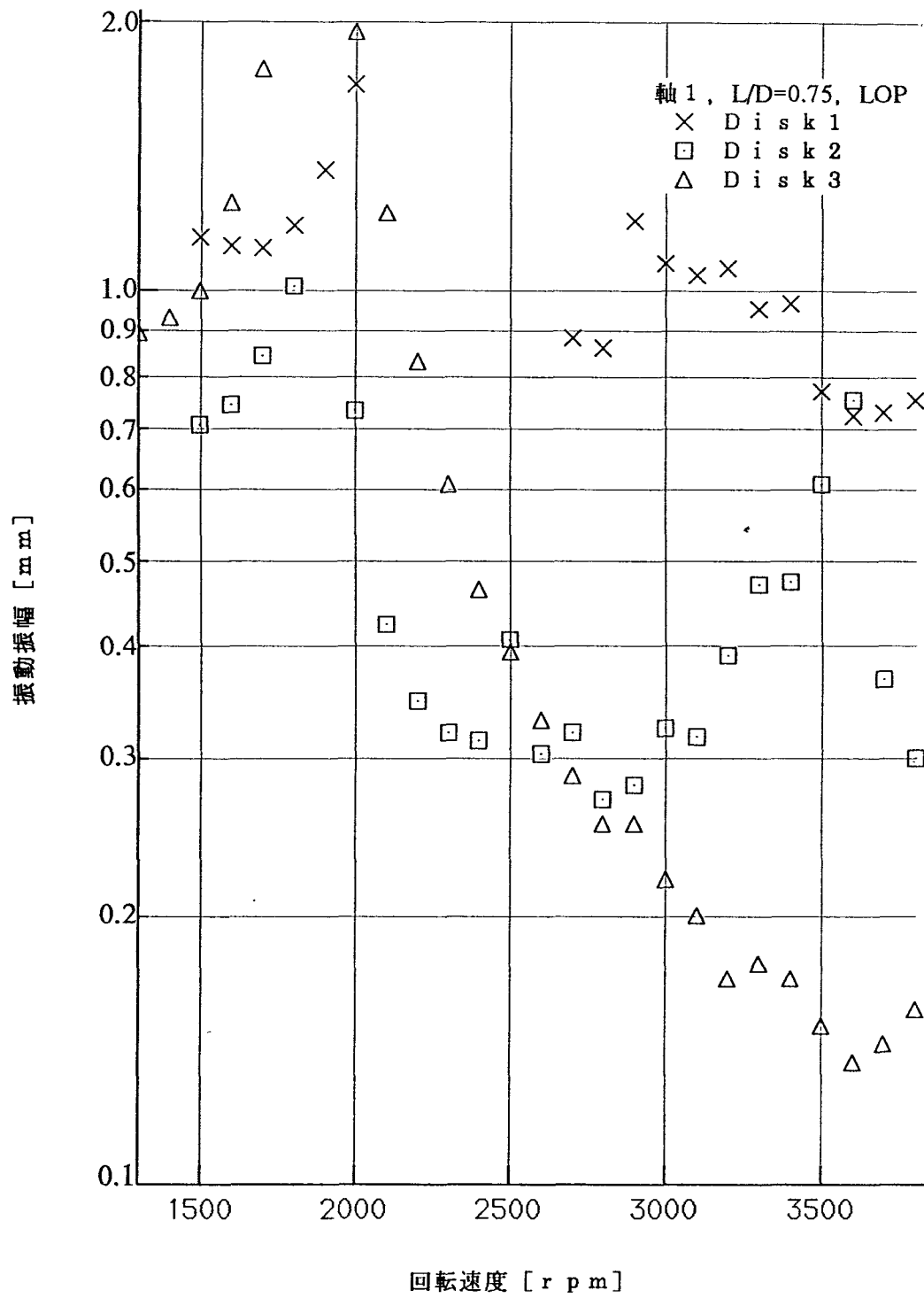
実験結果（本文に載せた結果以外）



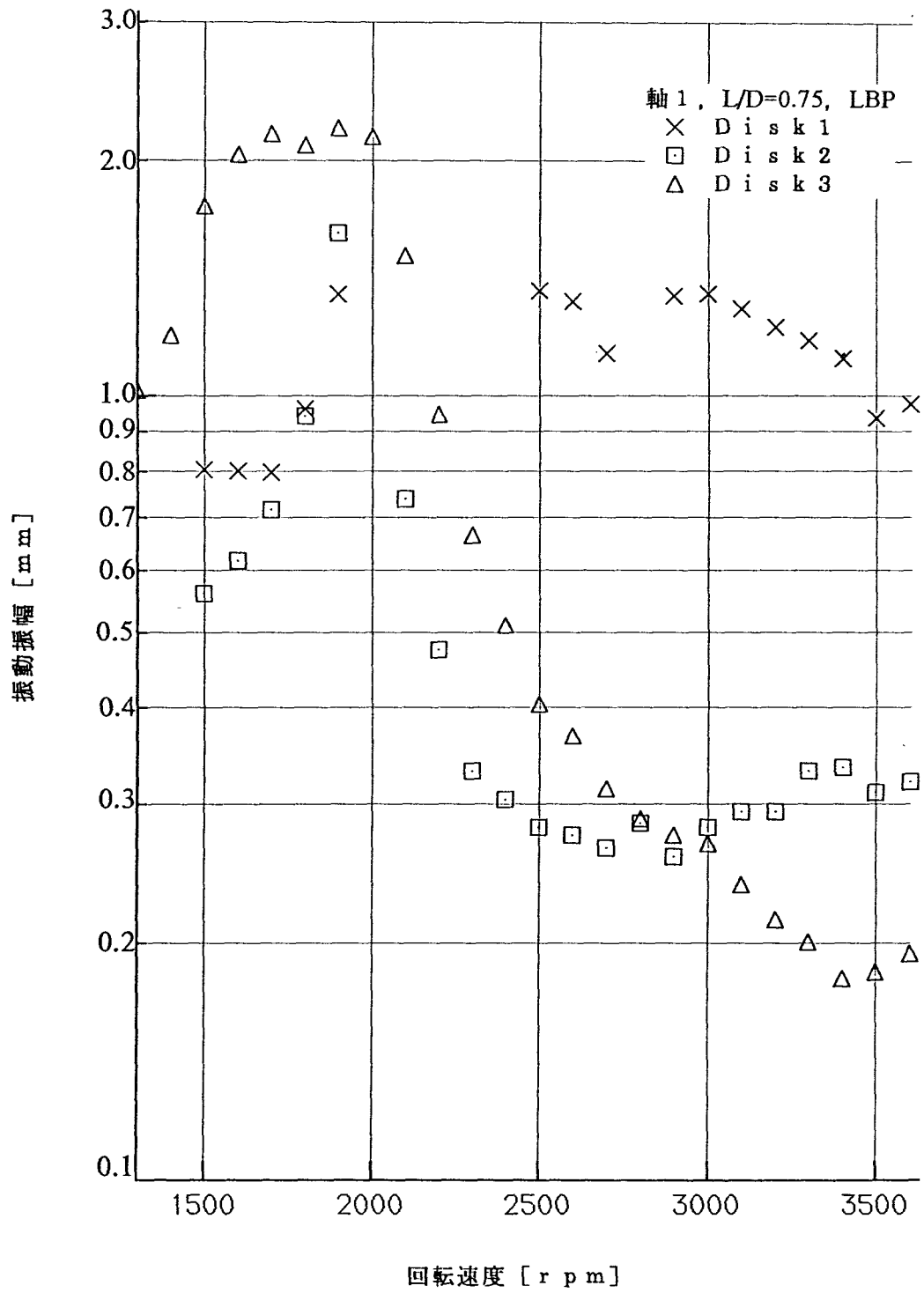
実験結果（本文に載せた結果以外）



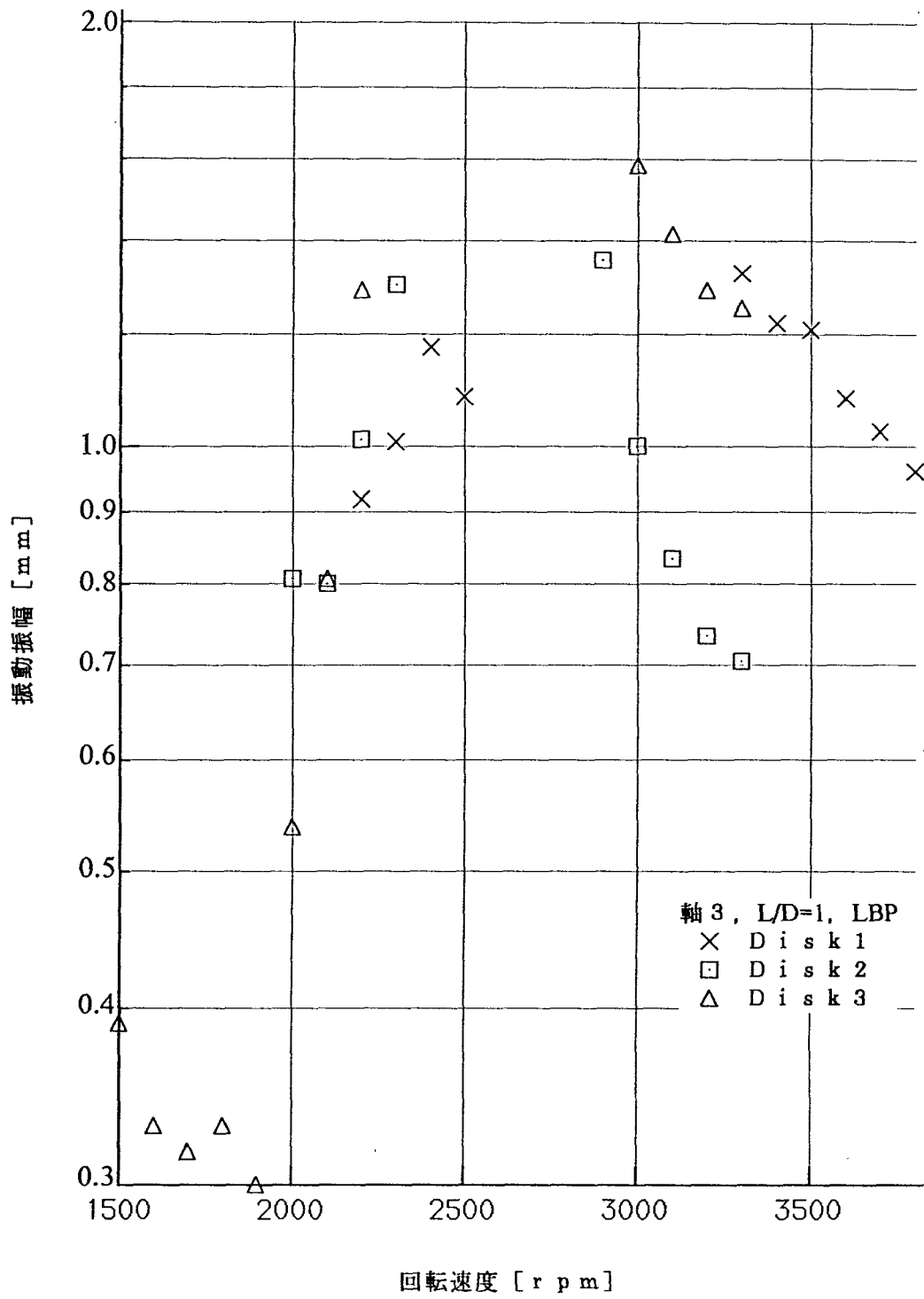
実験結果（本文に載せた結果以外）



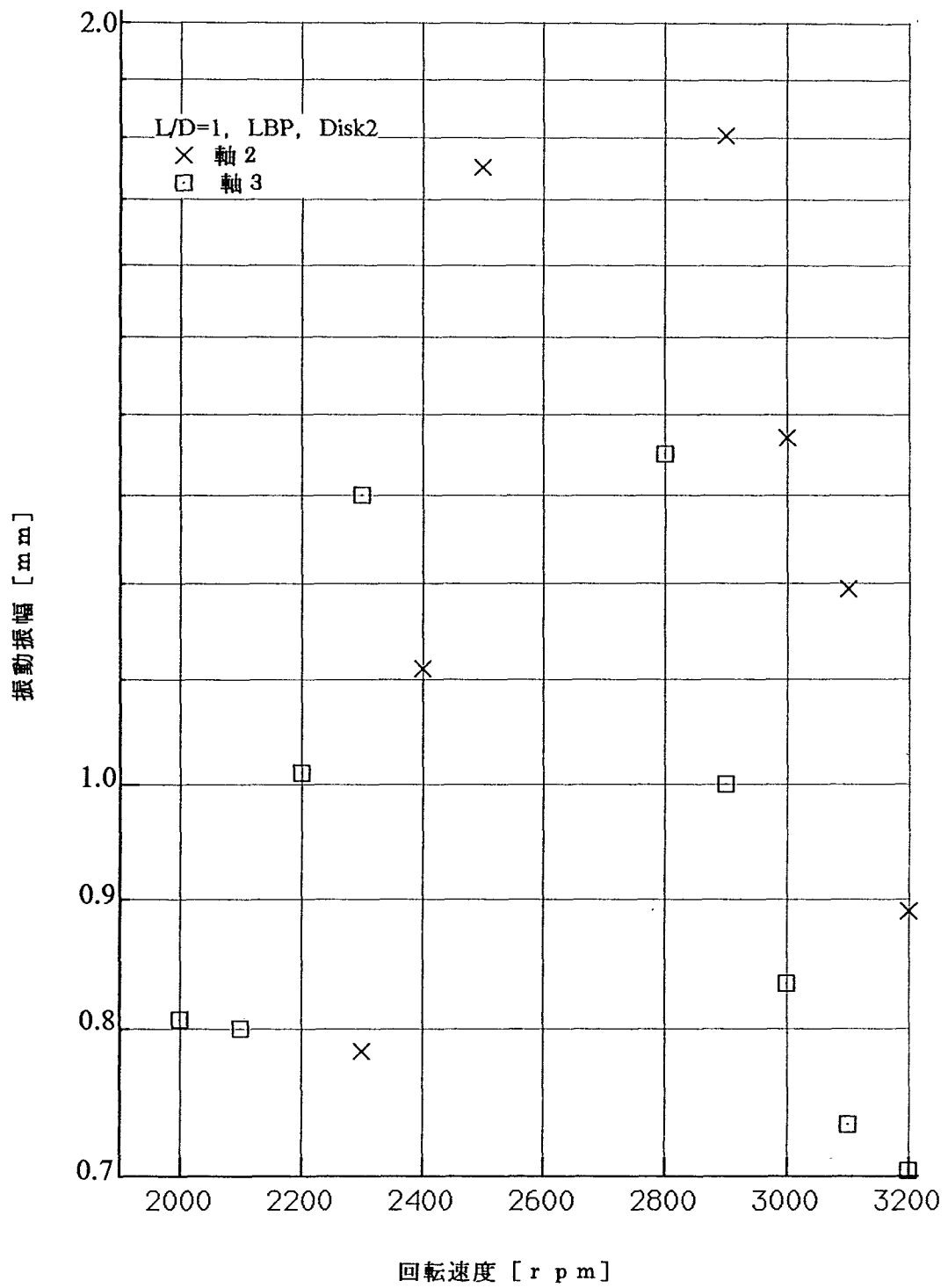
実験結果 (本文に載せた結果以外)



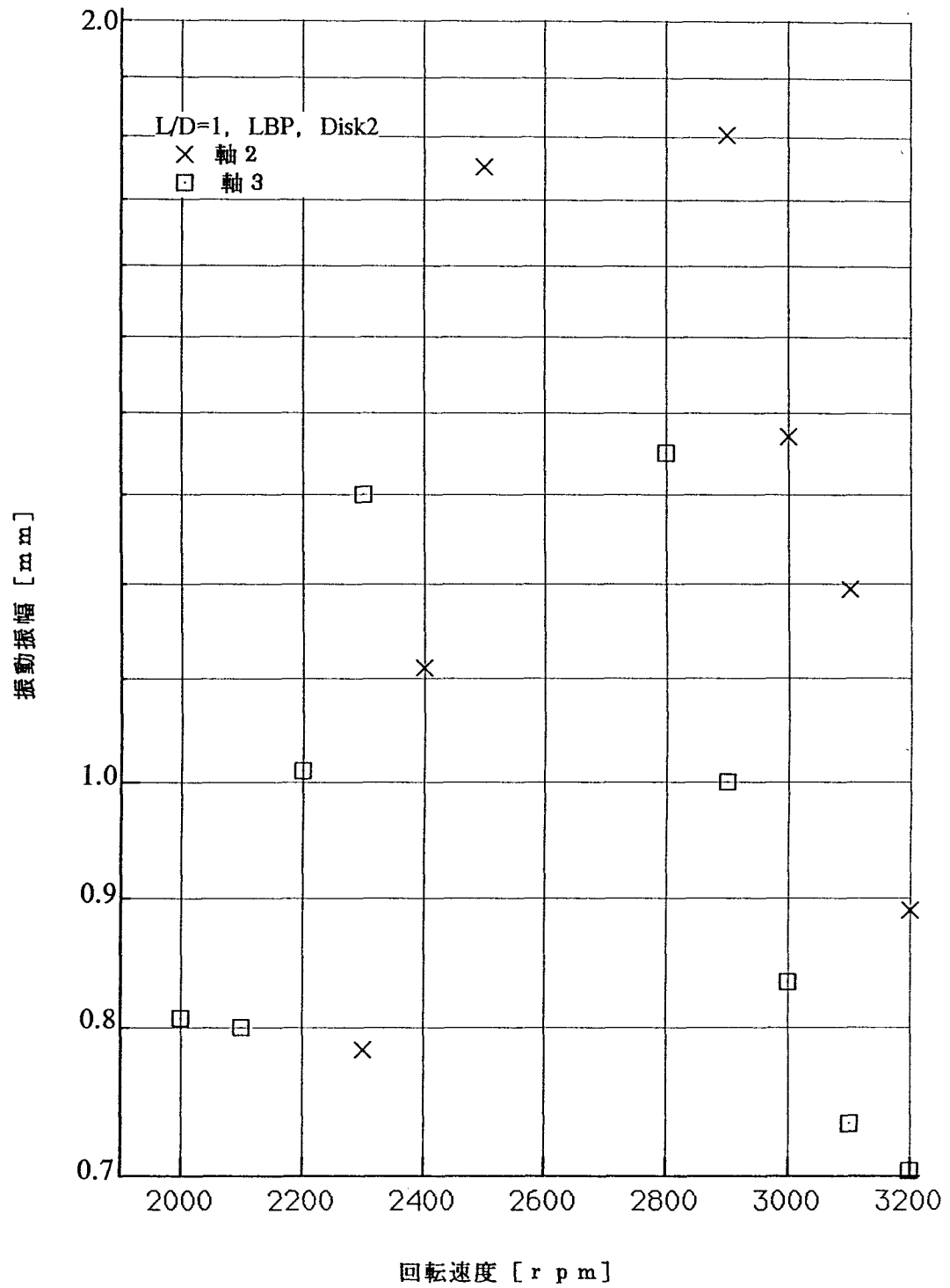
実験結果 (本文に載せた結果以外)



実験結果 (本文に載せた結果以外)

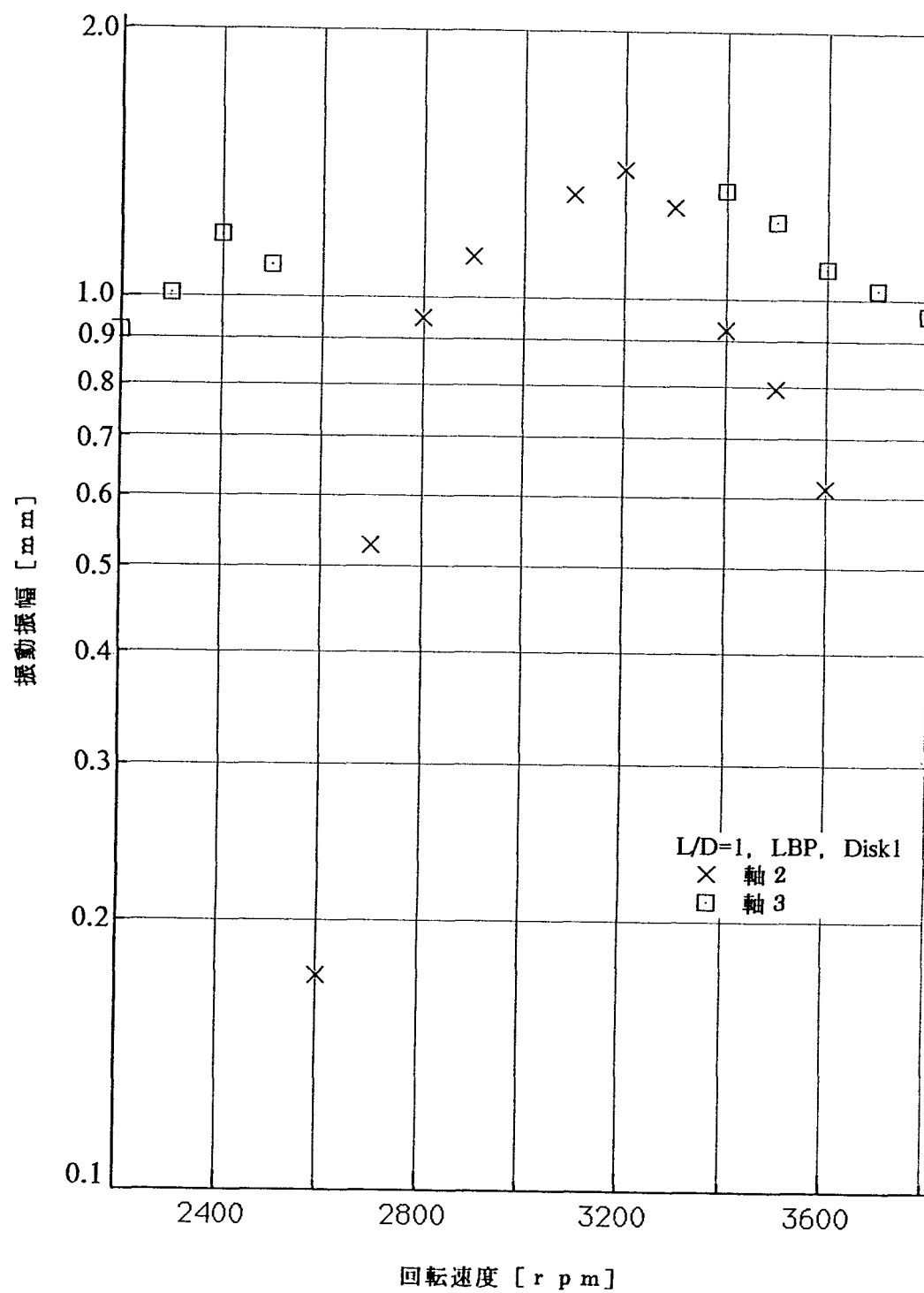


実験結果 (本文に載せた結果以外)

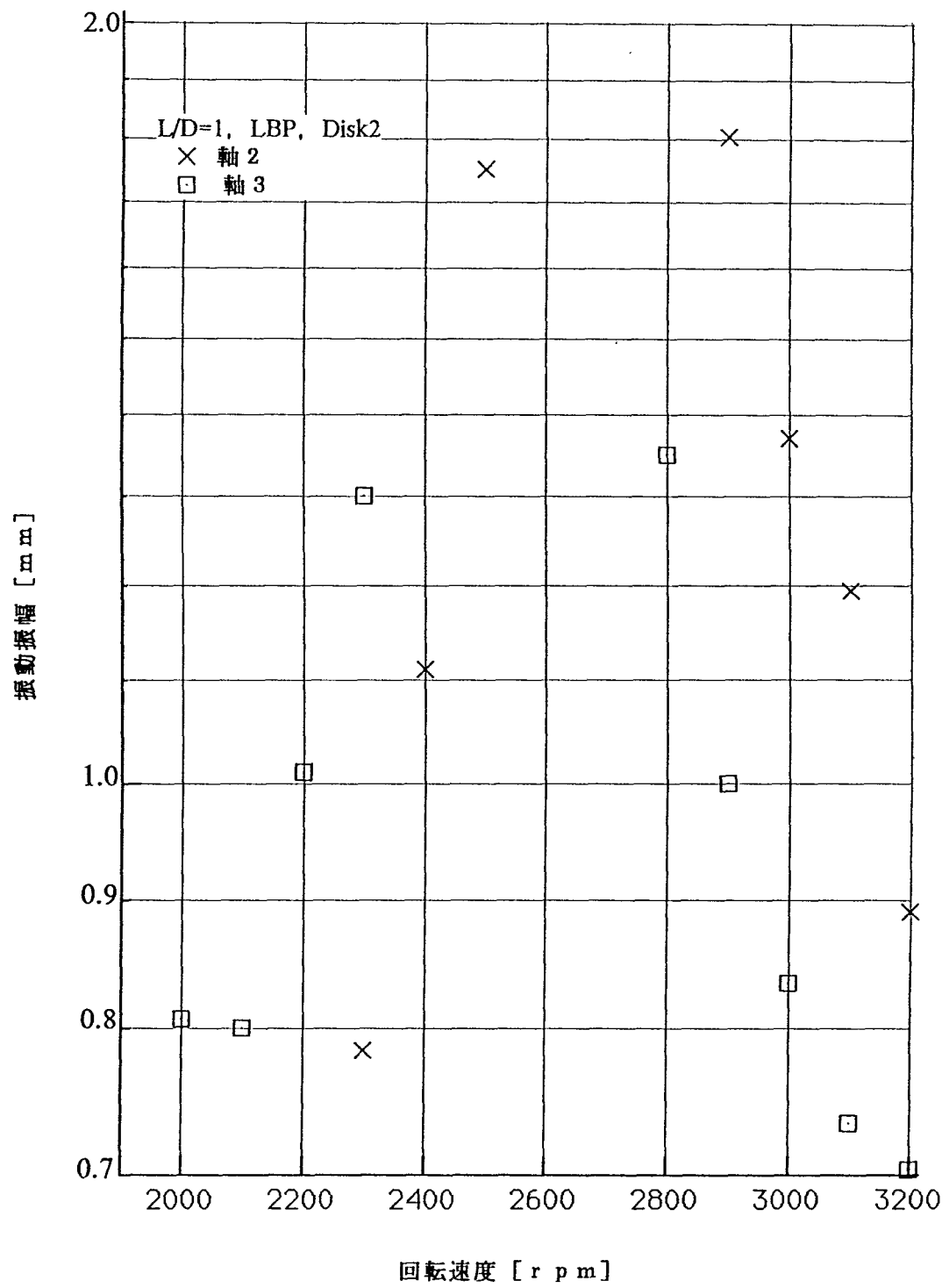


実験結果 (本文に載せた結果以外)

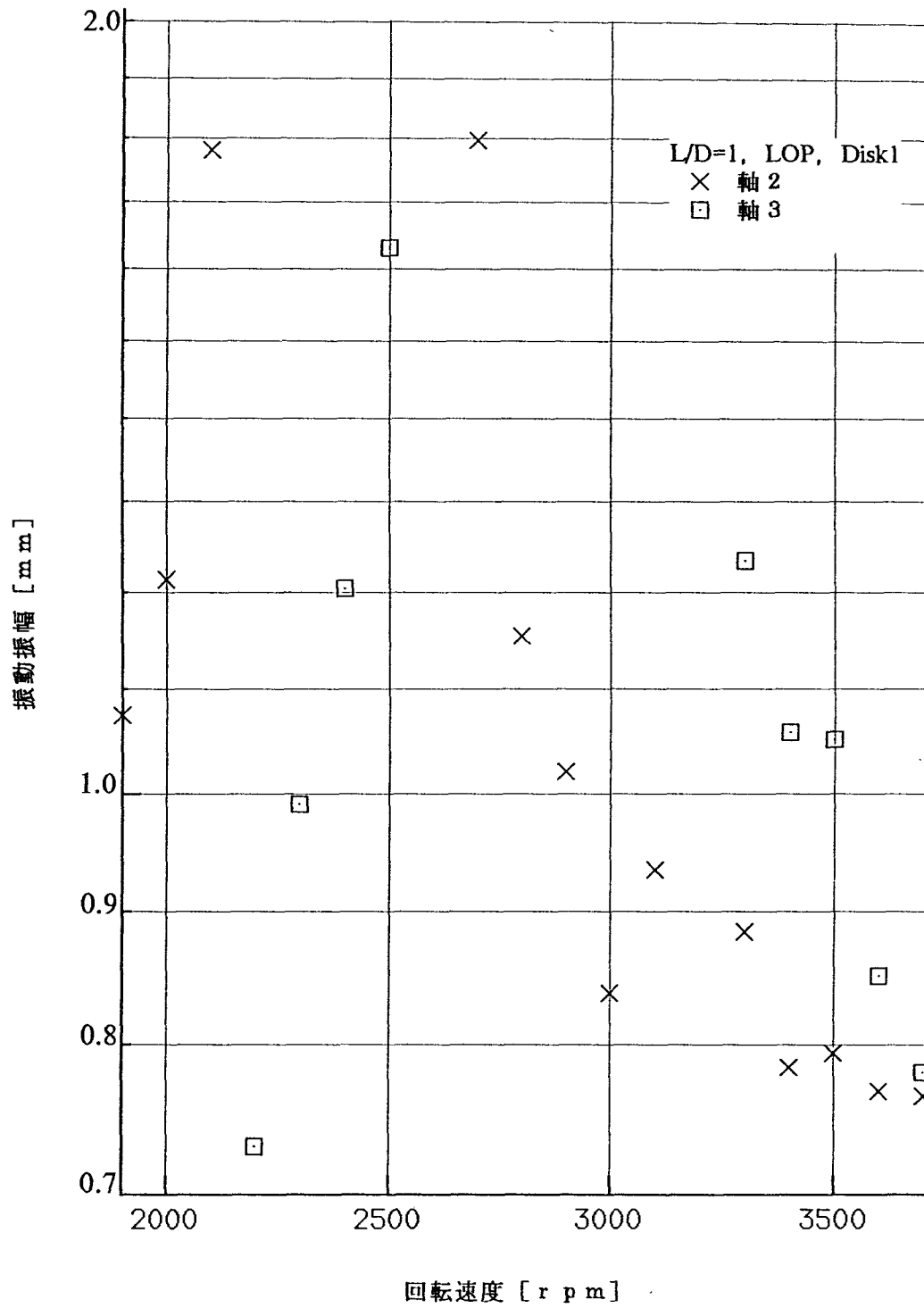




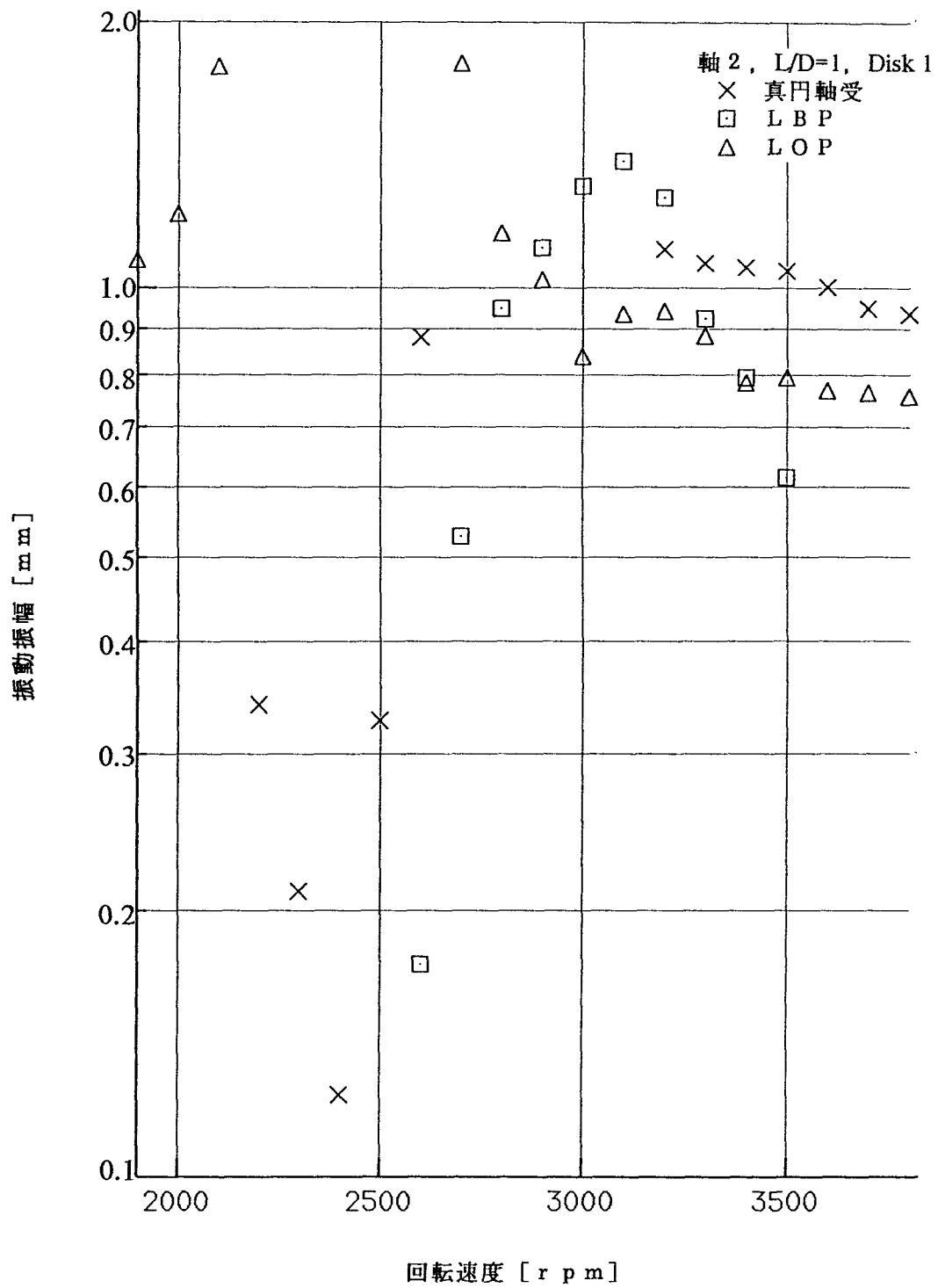
実験結果 (本文に載せた結果以外)



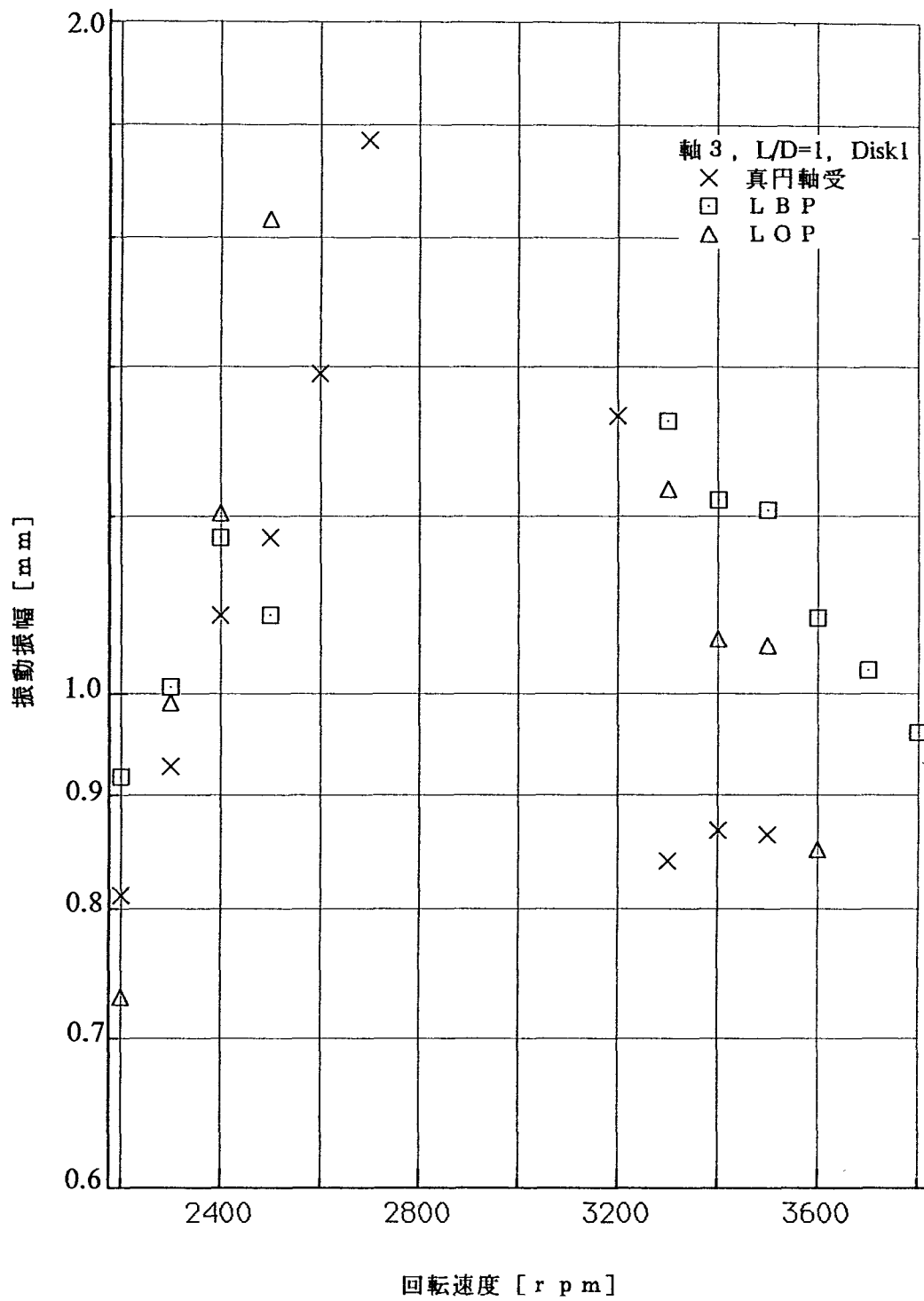
実験結果 (本文に載せた結果以外)



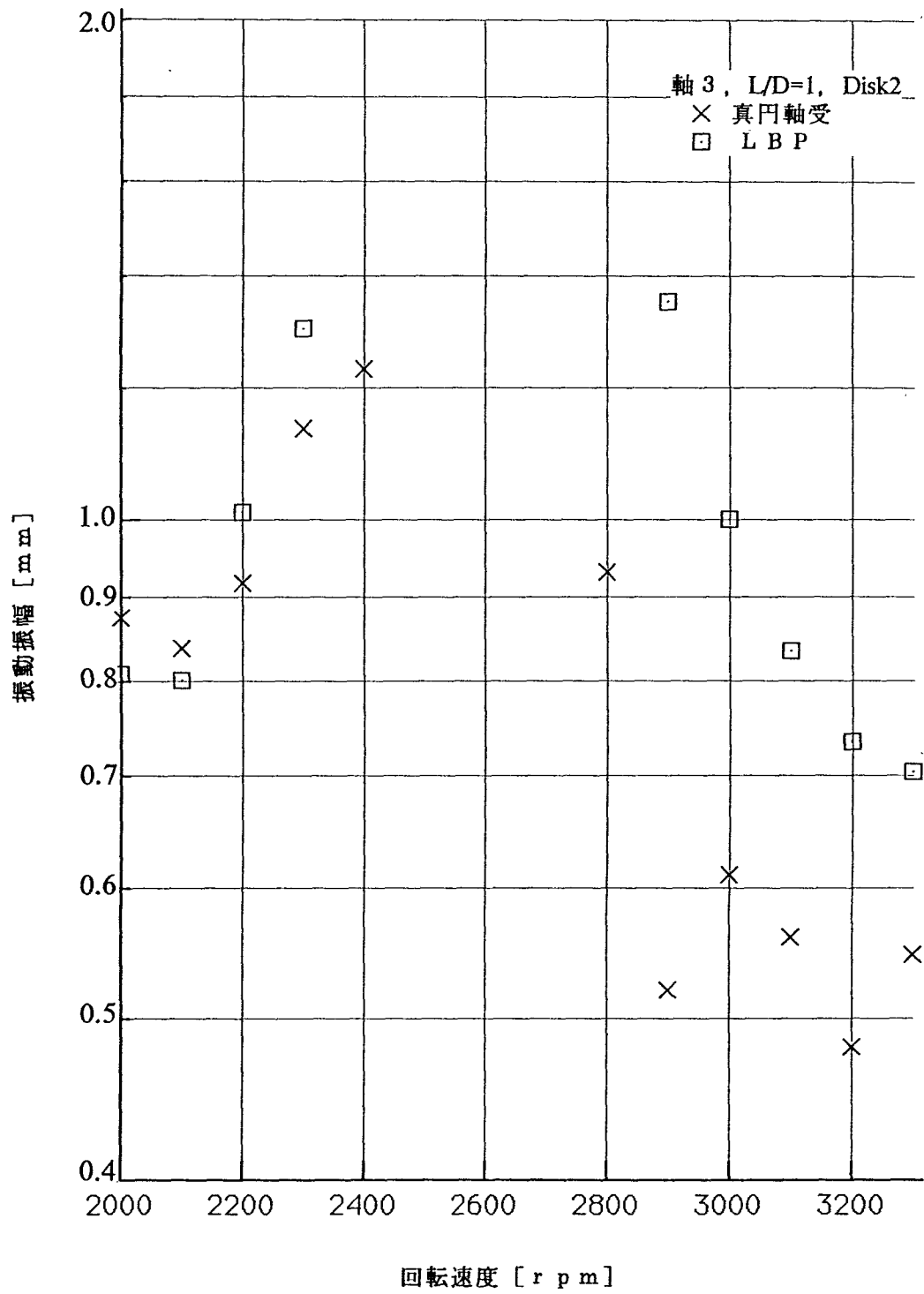
実験結果 (本文に載せた結果以外)



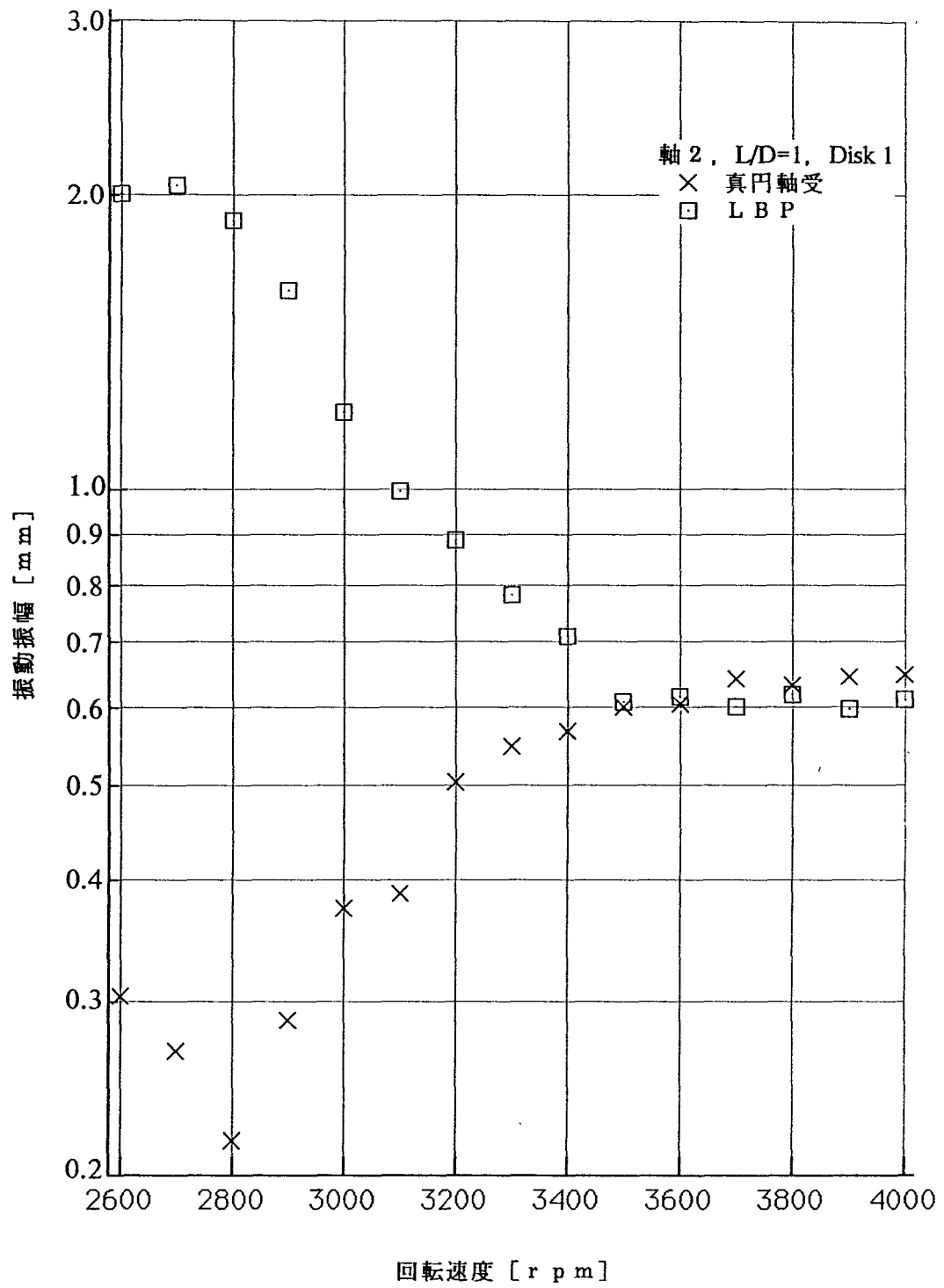
実験結果 (本文に載せた結果以外)



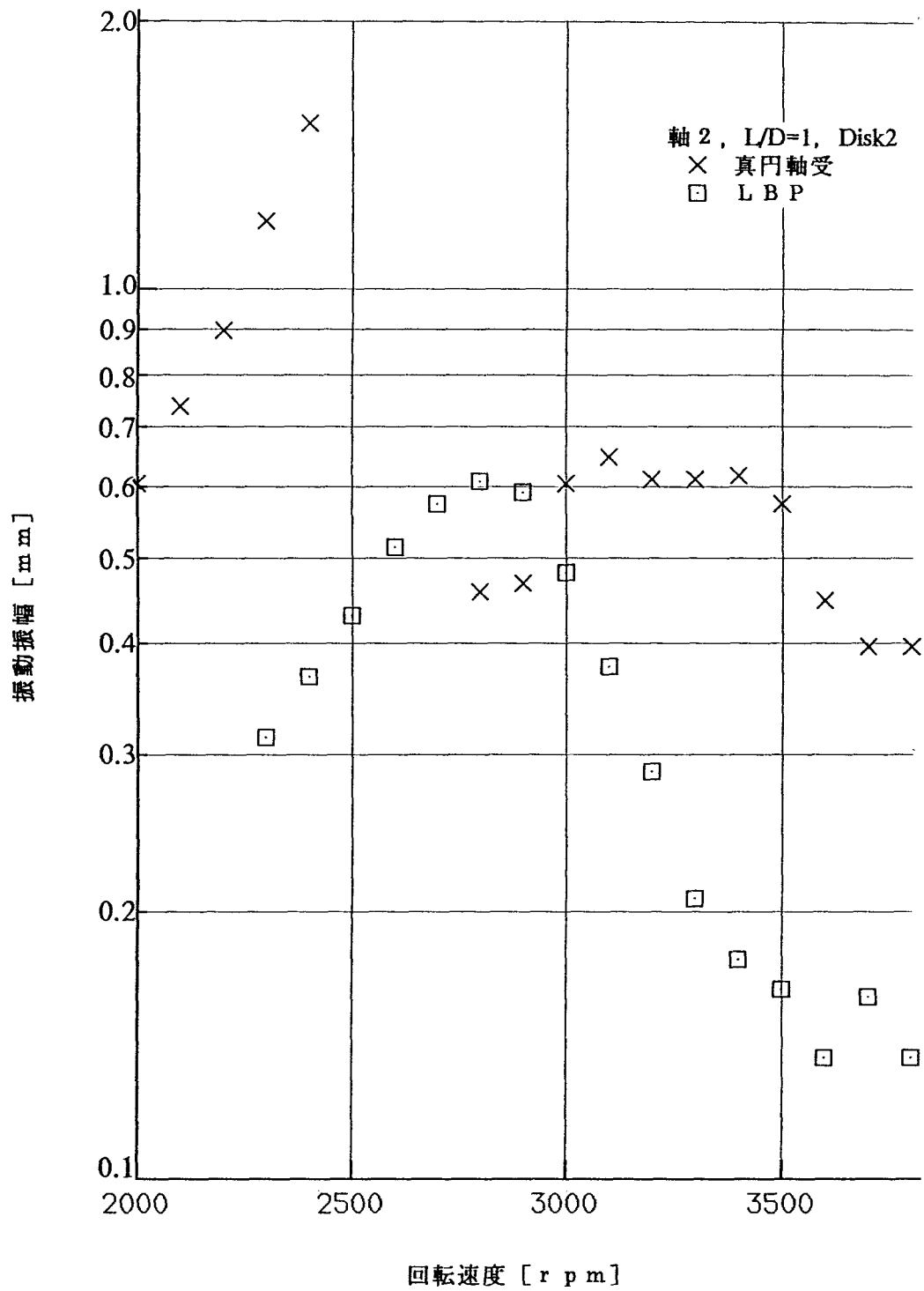
実験結果 (本文に載せた結果以外)



実験結果 (本文に載せた結果以外)

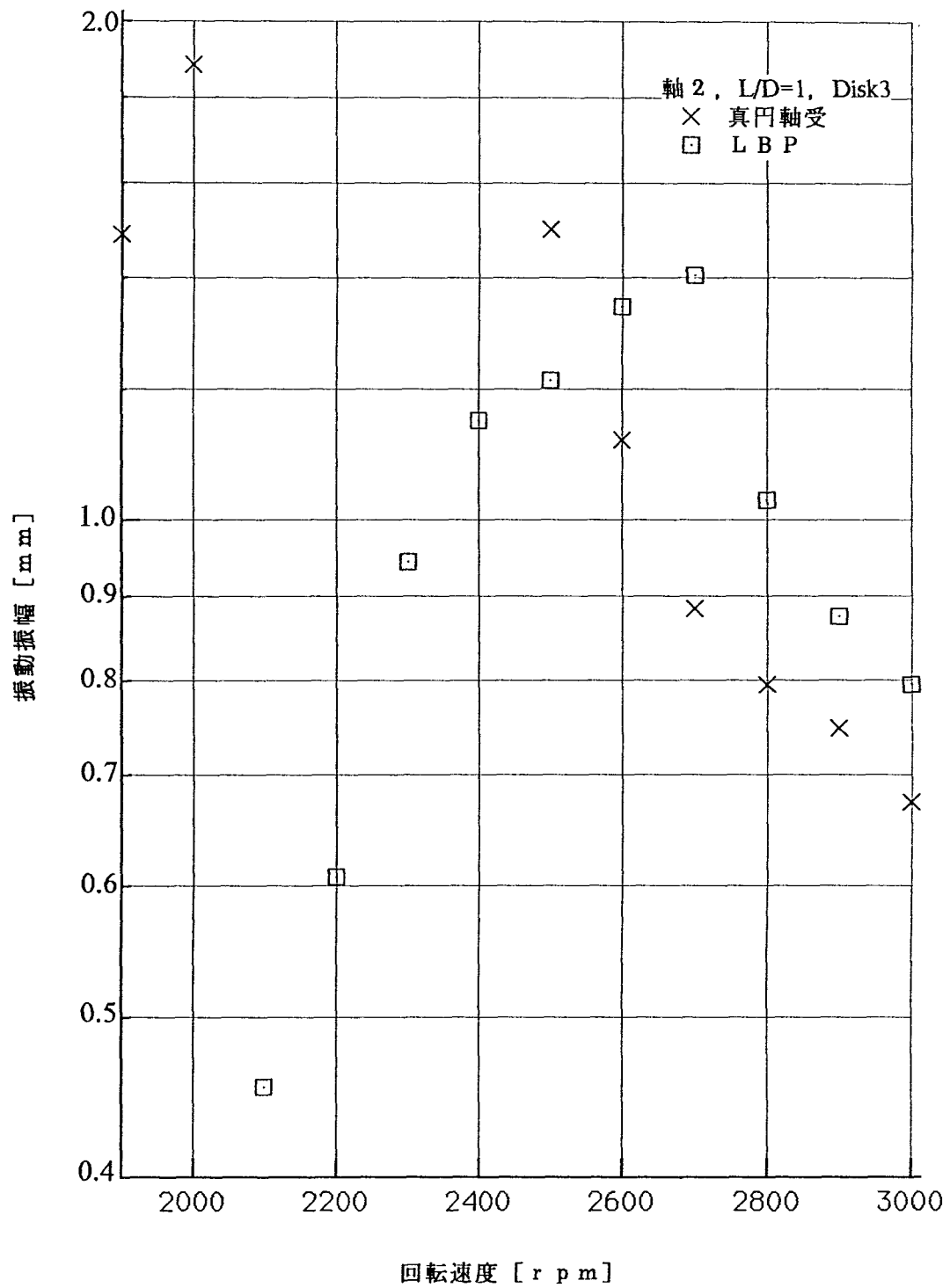


実験結果 (本文に載せた結果以外)

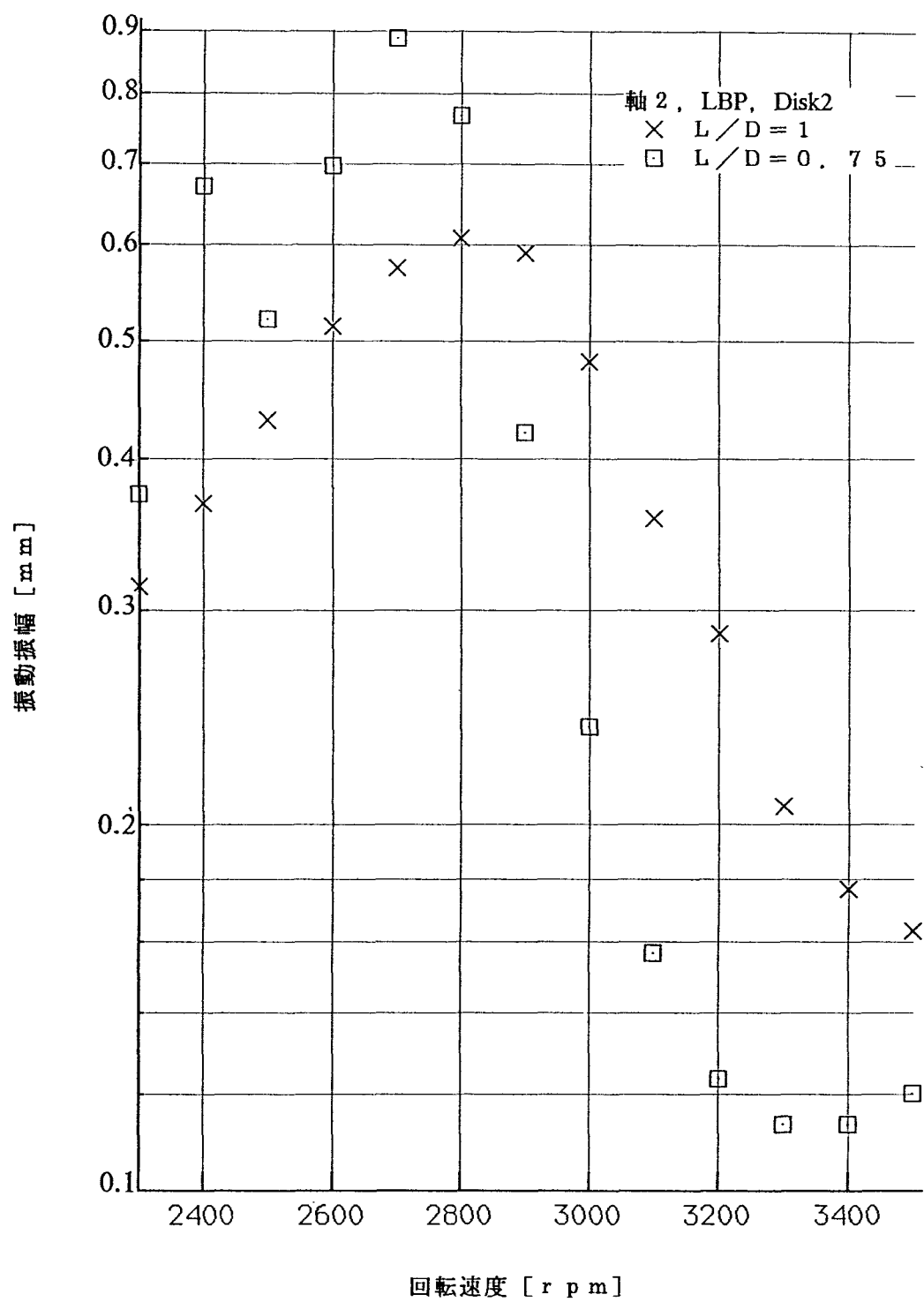


実験結果 (本文に載せた結果以外)

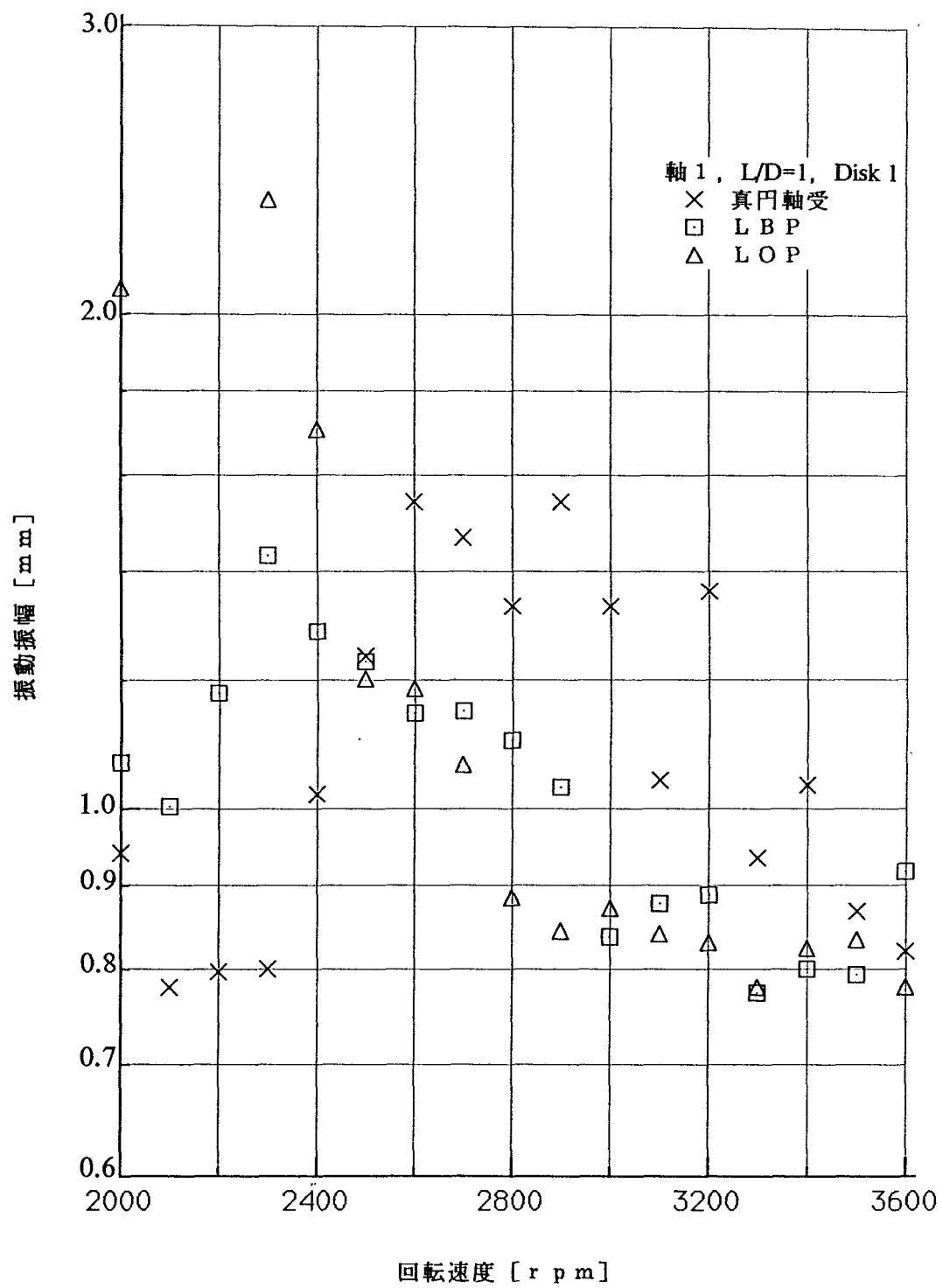




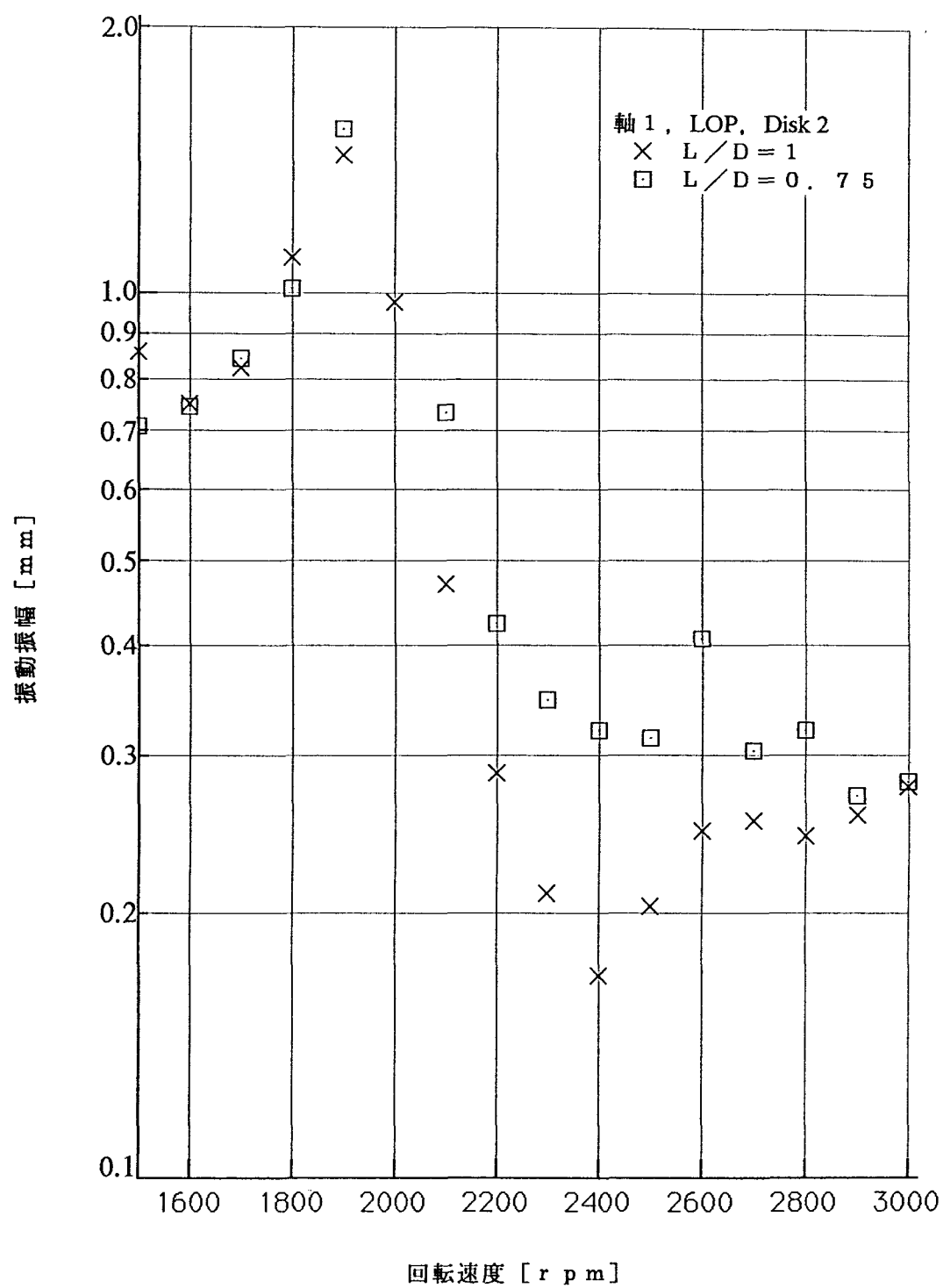
実験結果 (本文に載せた結果以外)



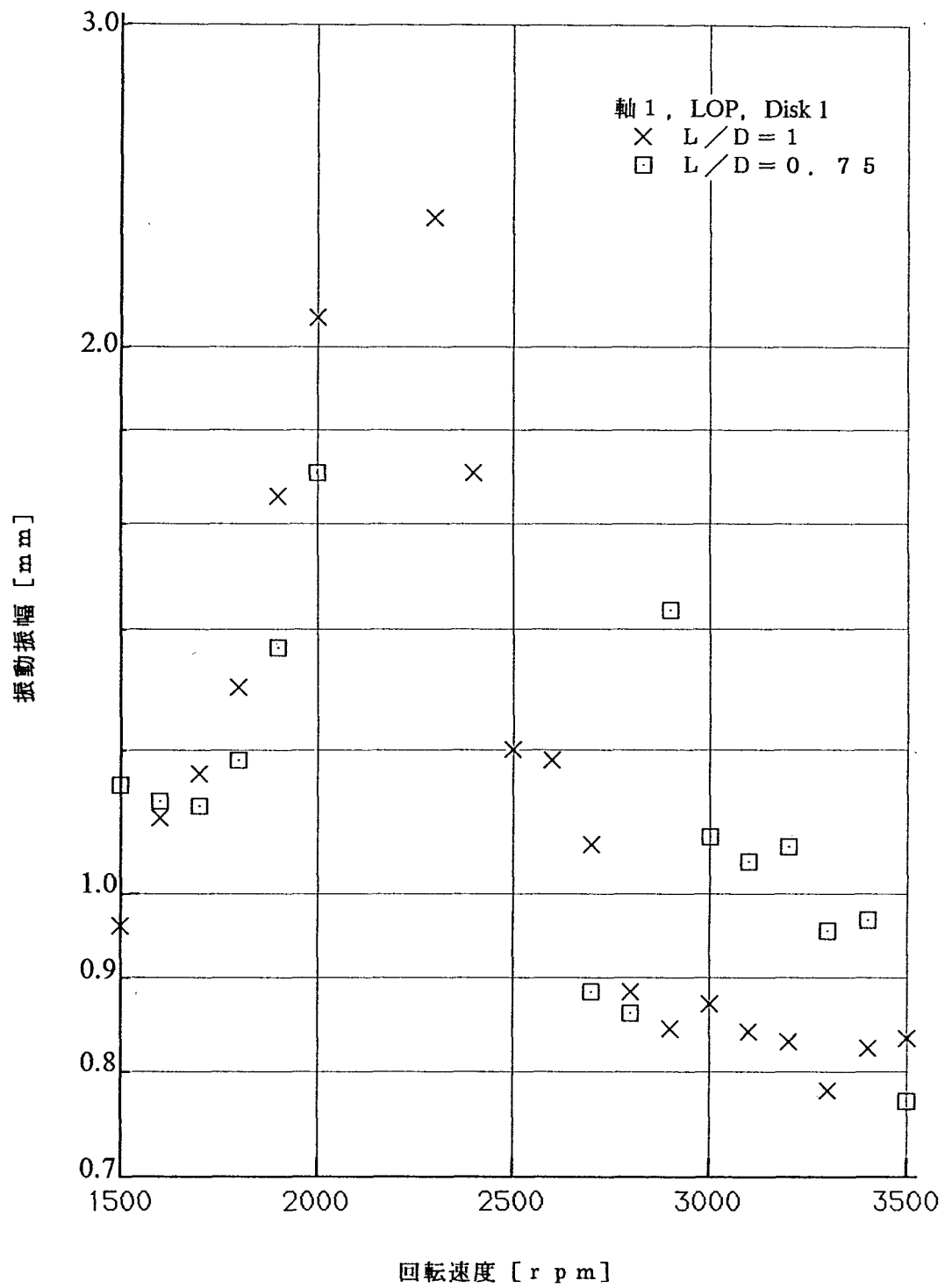
実験結果 (本文に載せた結果以外)



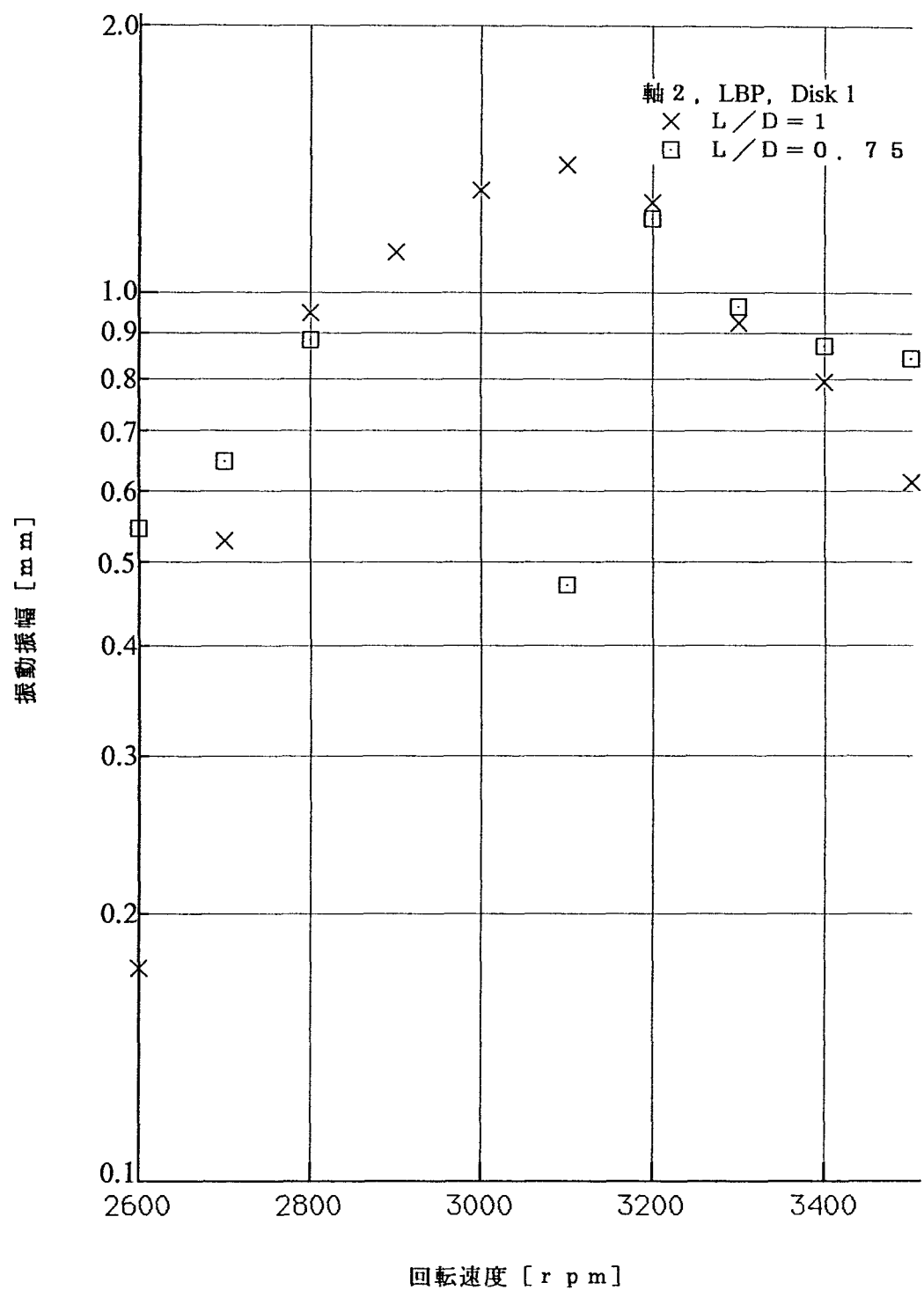
実験結果 (本文に載せた結果以外)



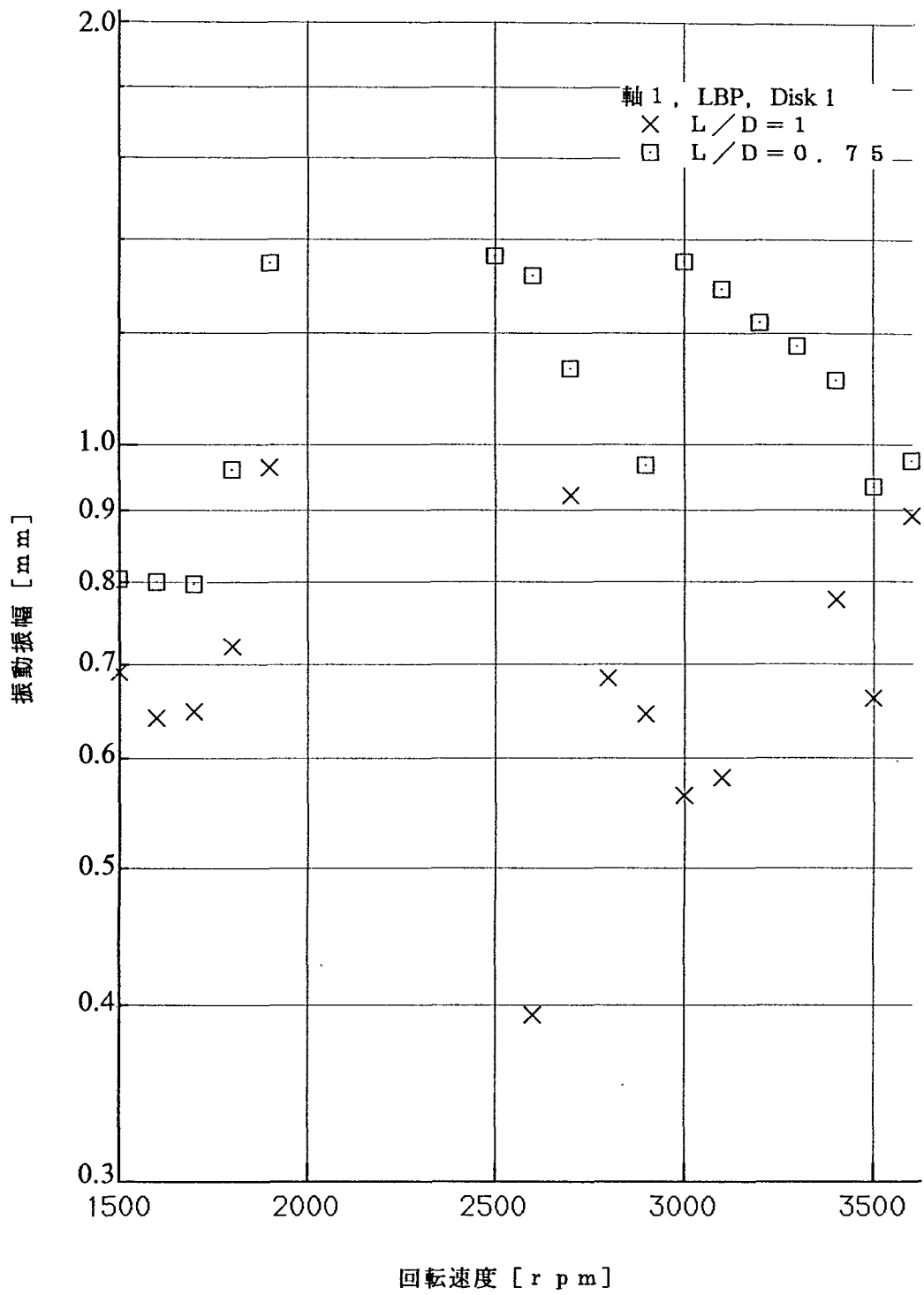
実験結果（本文に載せた結果以外）



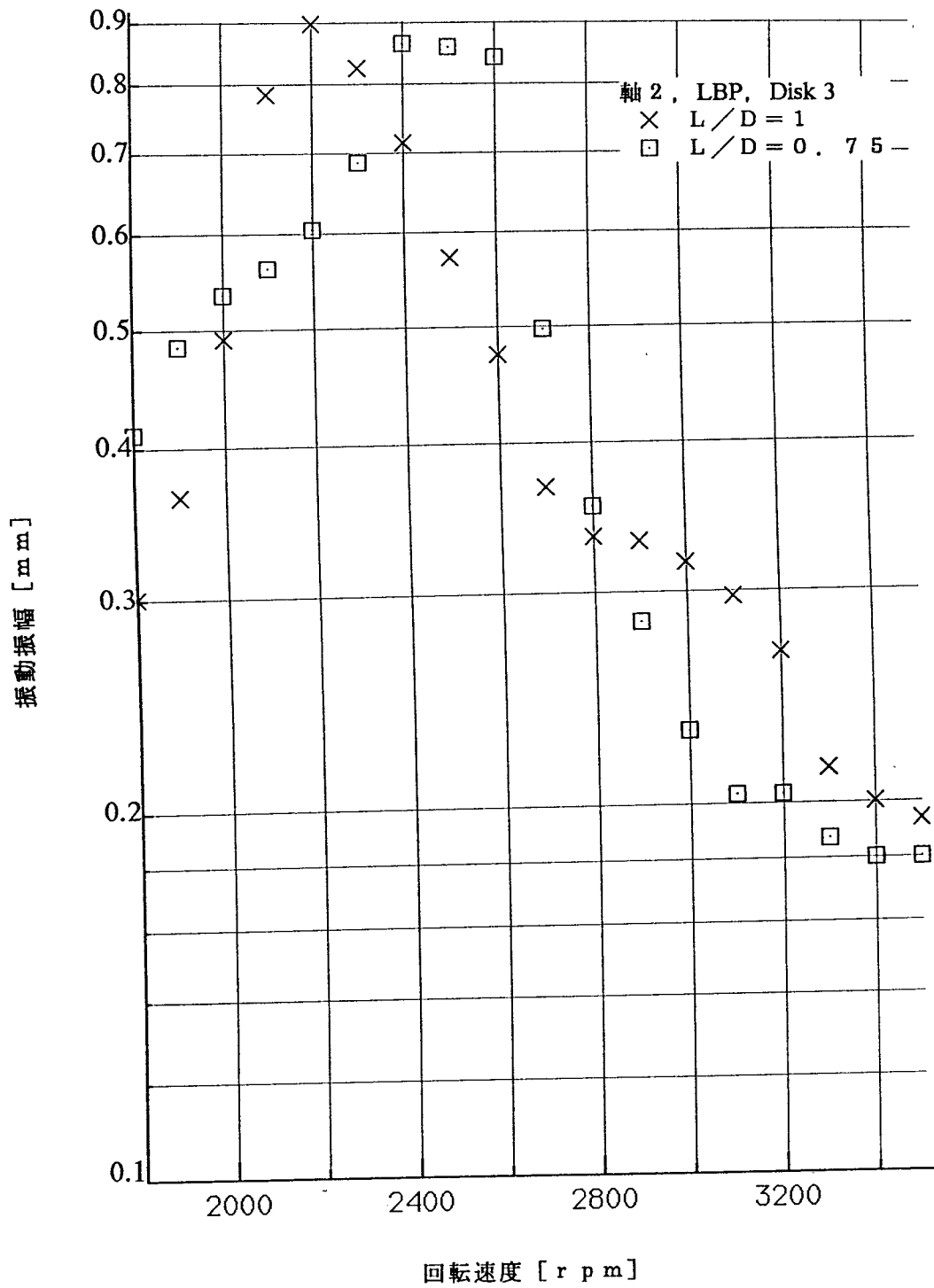
実験結果 (本文に載せた結果以外)



実験結果（本文に載せた結果以外）

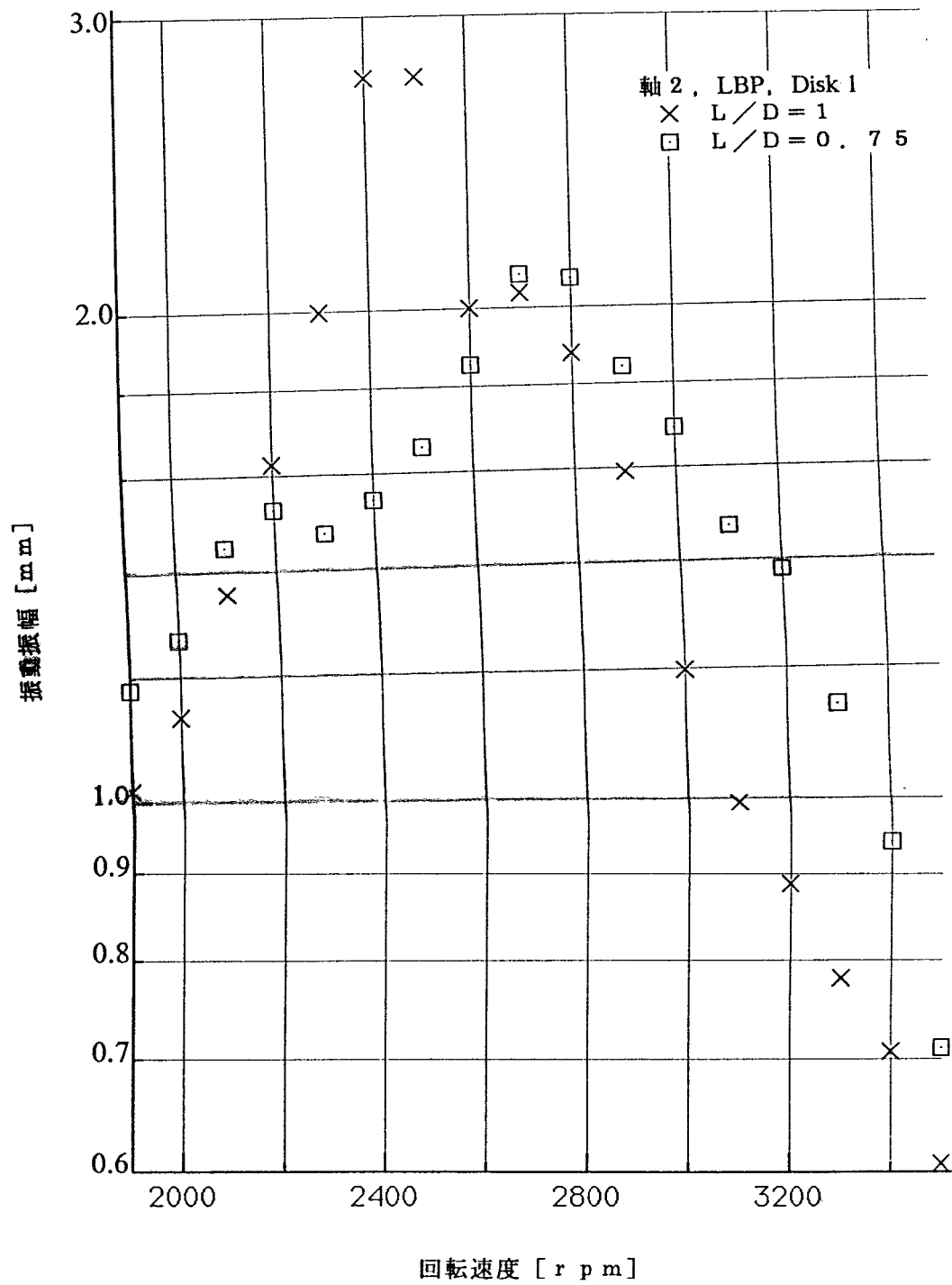


実験結果 (本文に載せた結果以外)

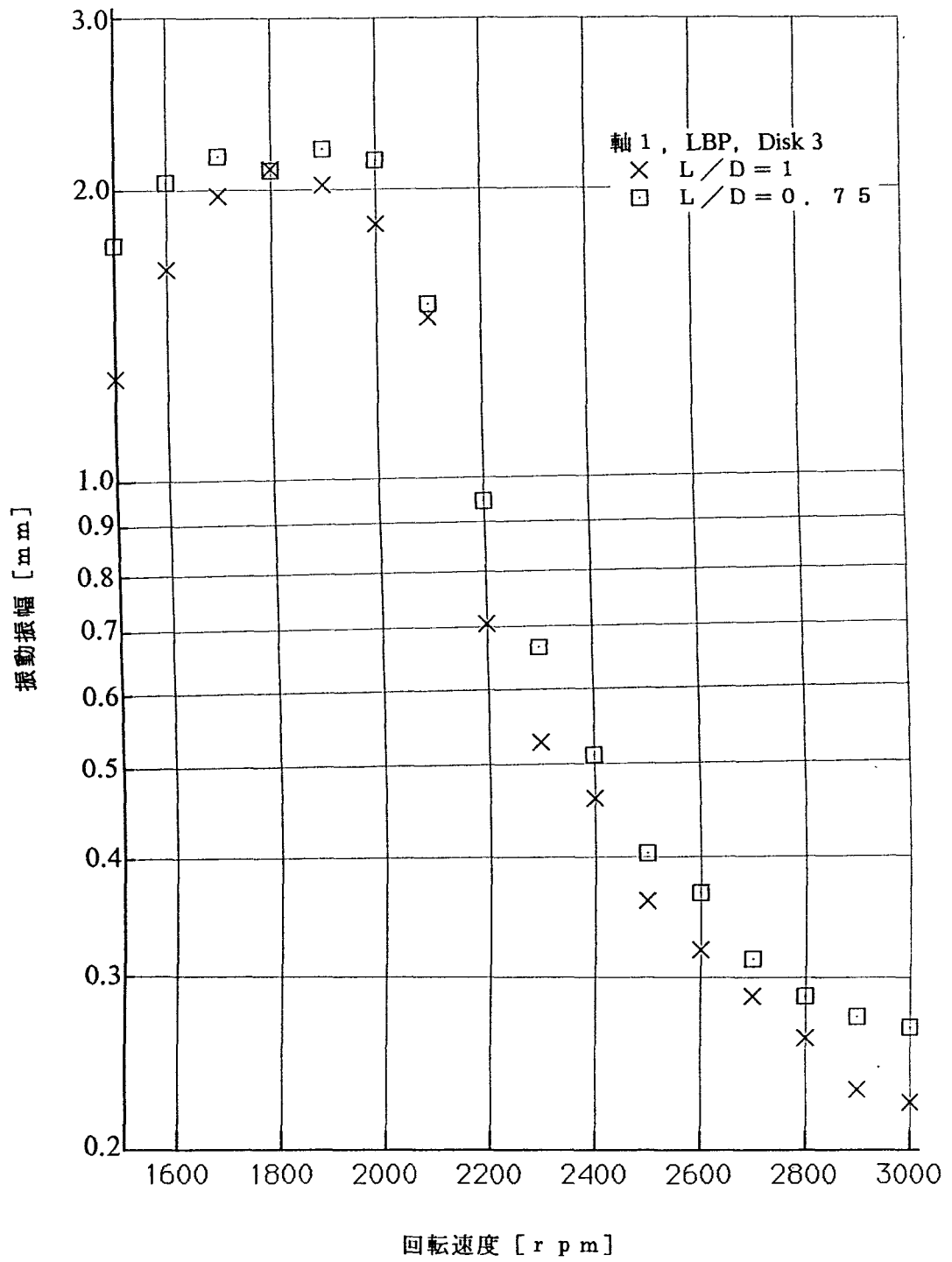


実験結果（本文に載せた結果以外）

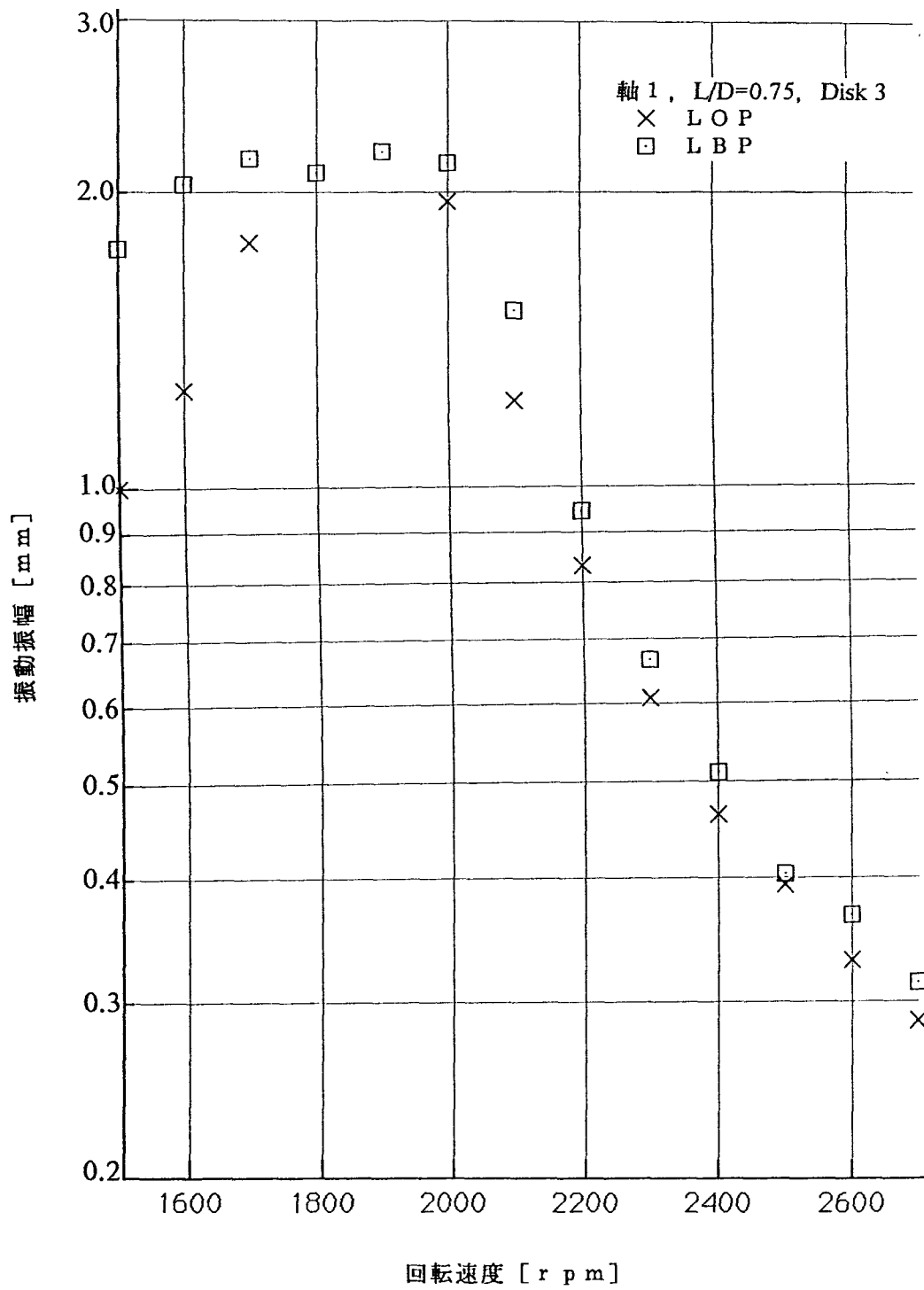




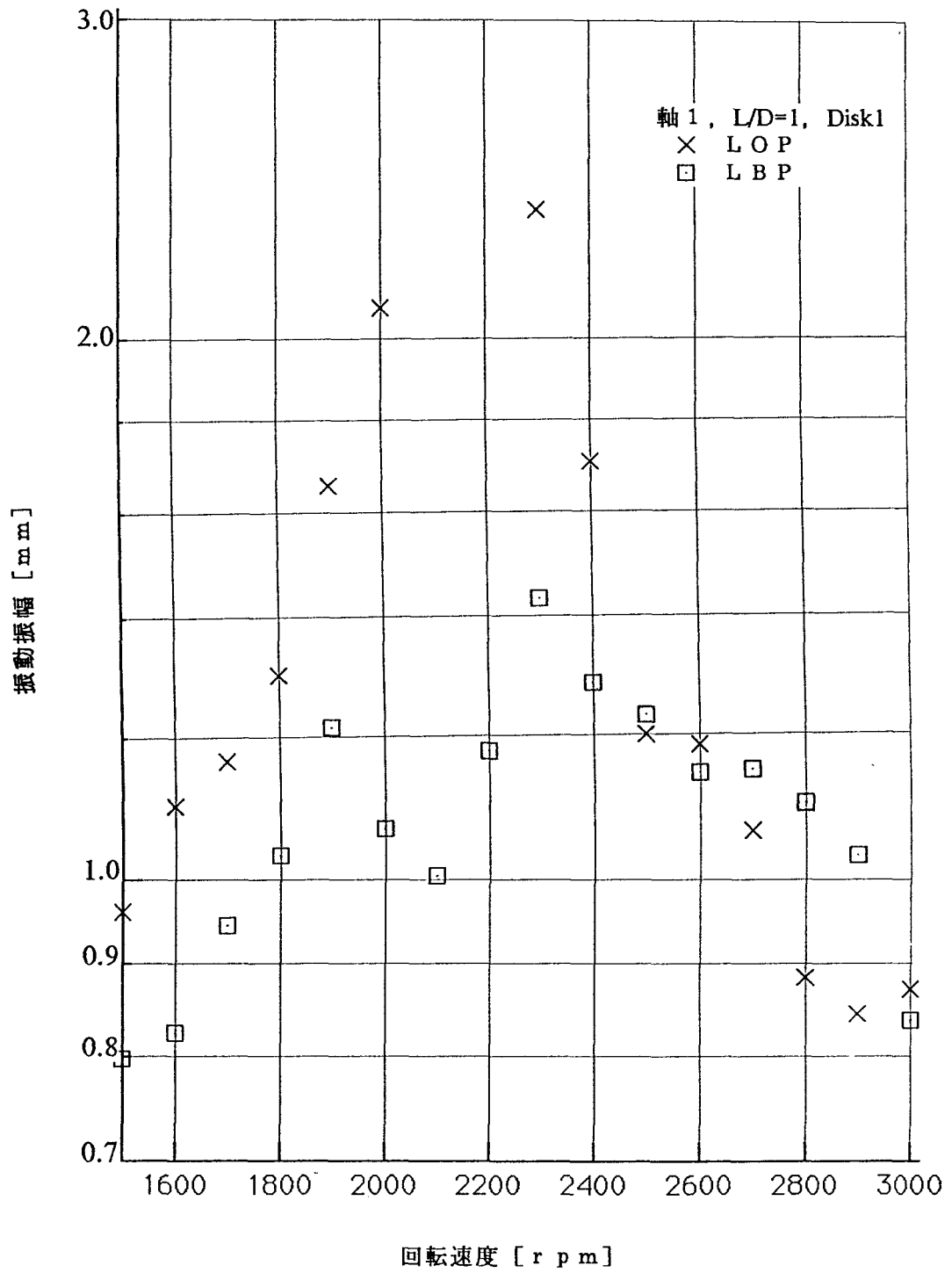
実験結果（本文に載せた結果以外）



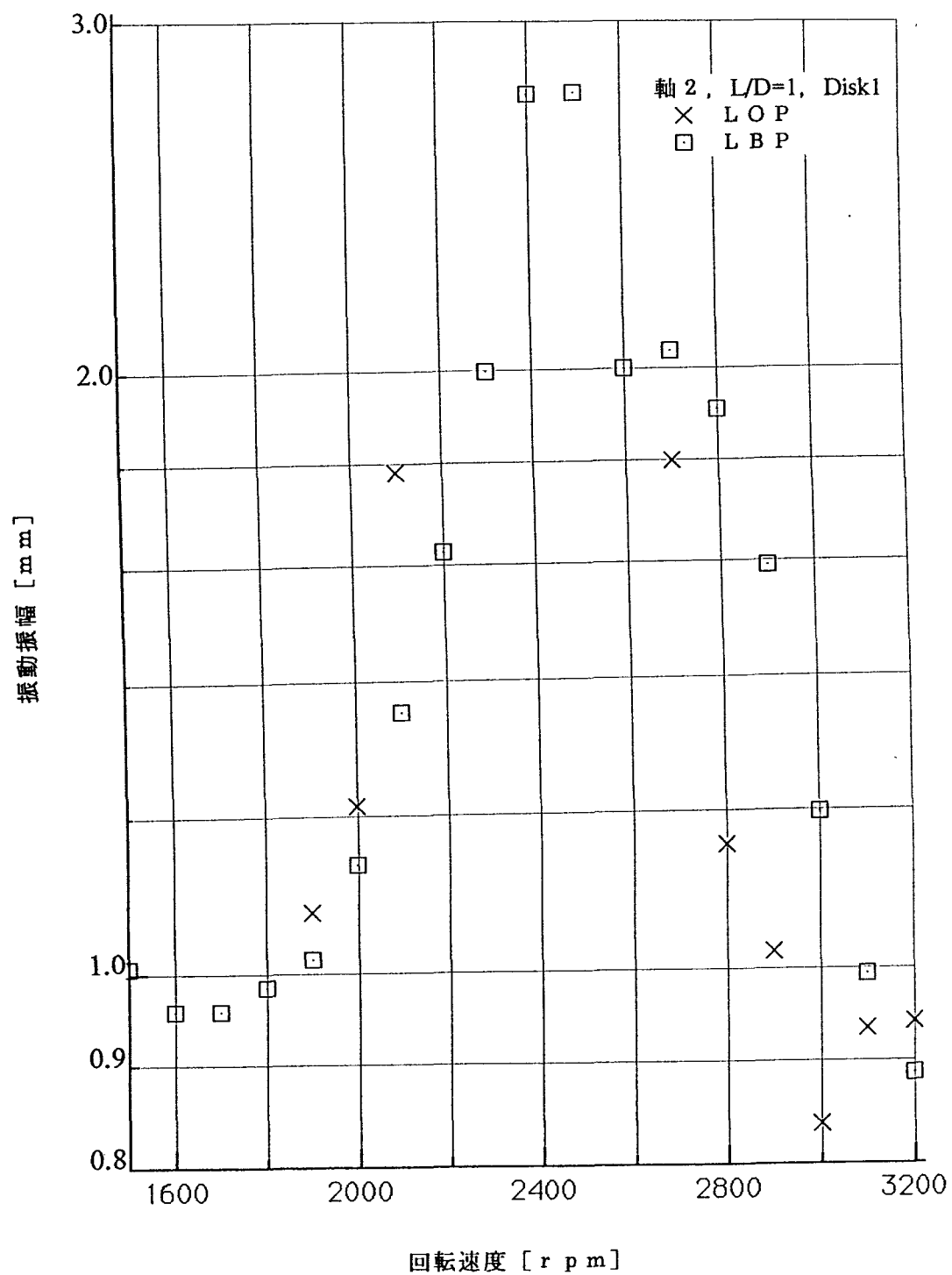
実験結果（本文に載せた結果以外）



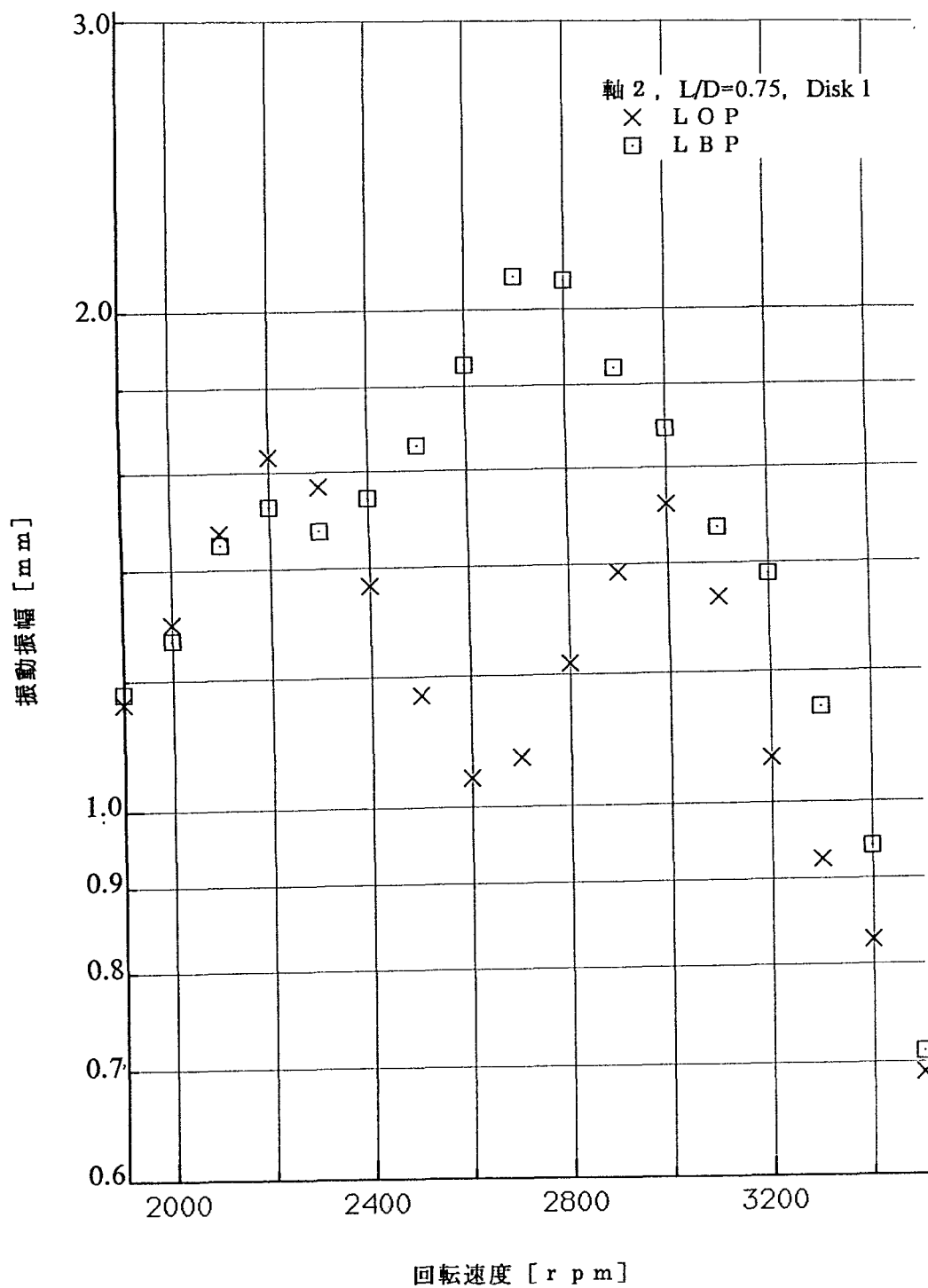
実験結果 (本文に載せた結果以外)



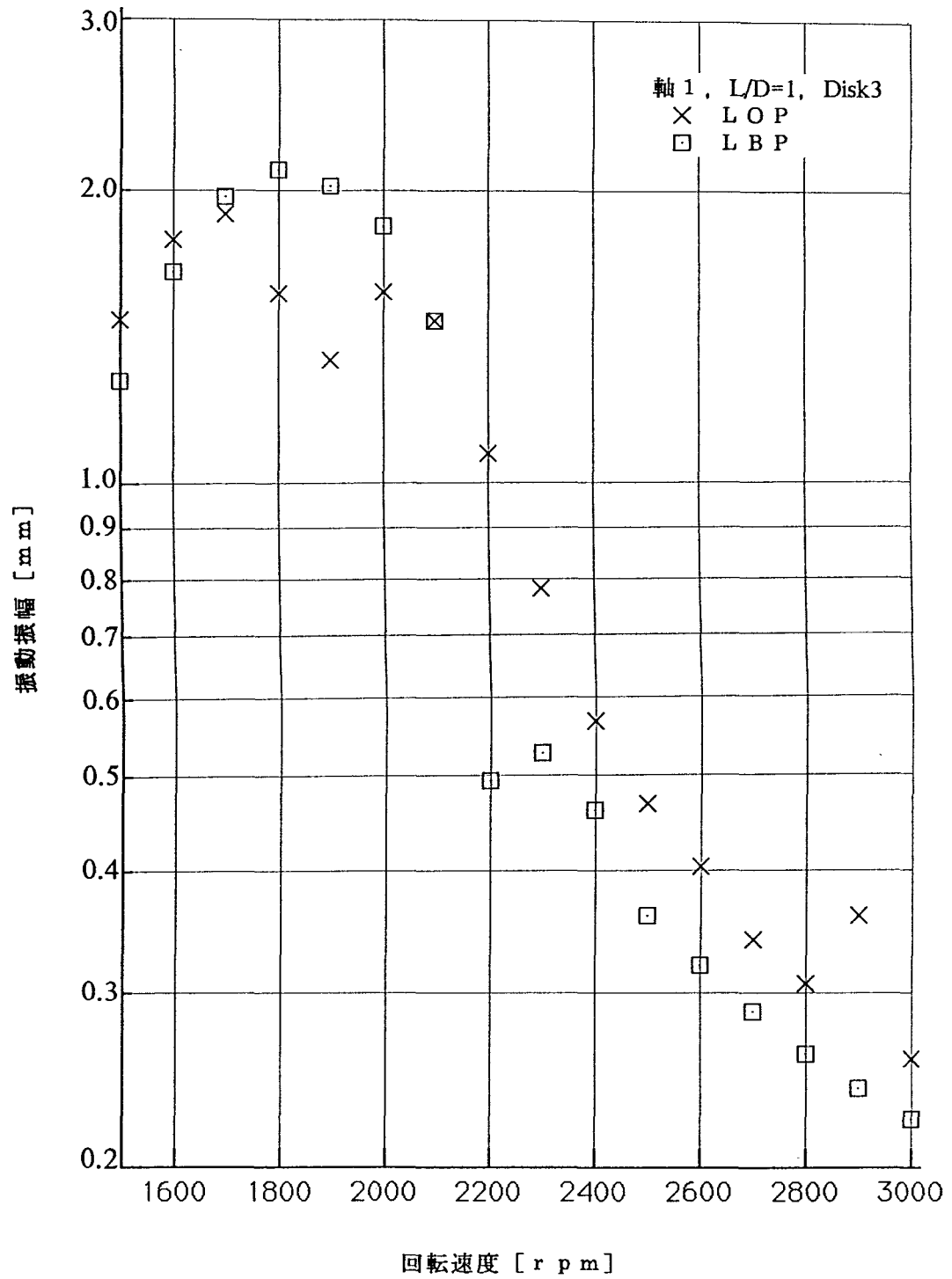
実験結果（本文に載せた結果以外）



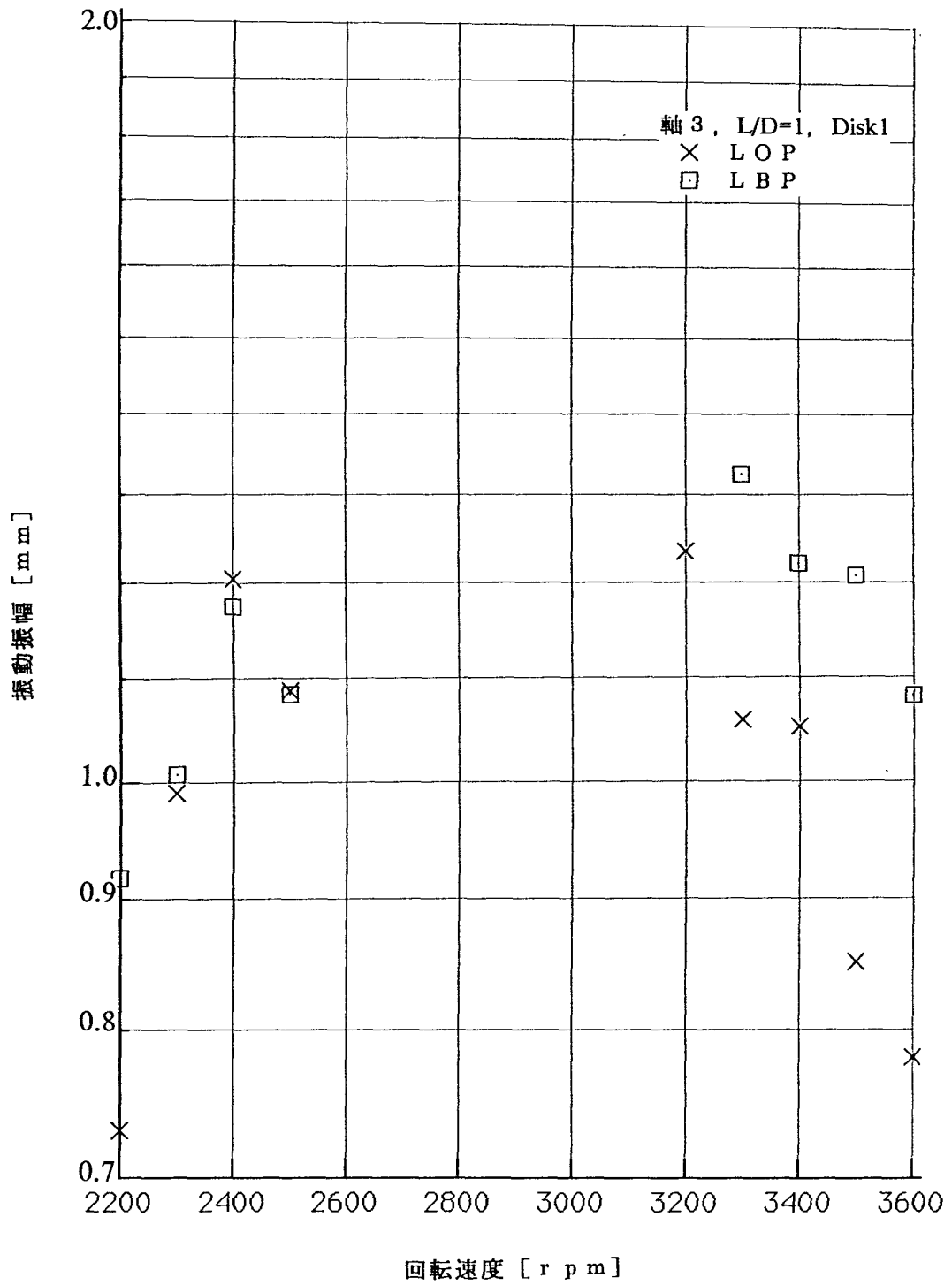
実験結果（本文に載せた結果以外）



実験結果（本文に載せた結果以外）

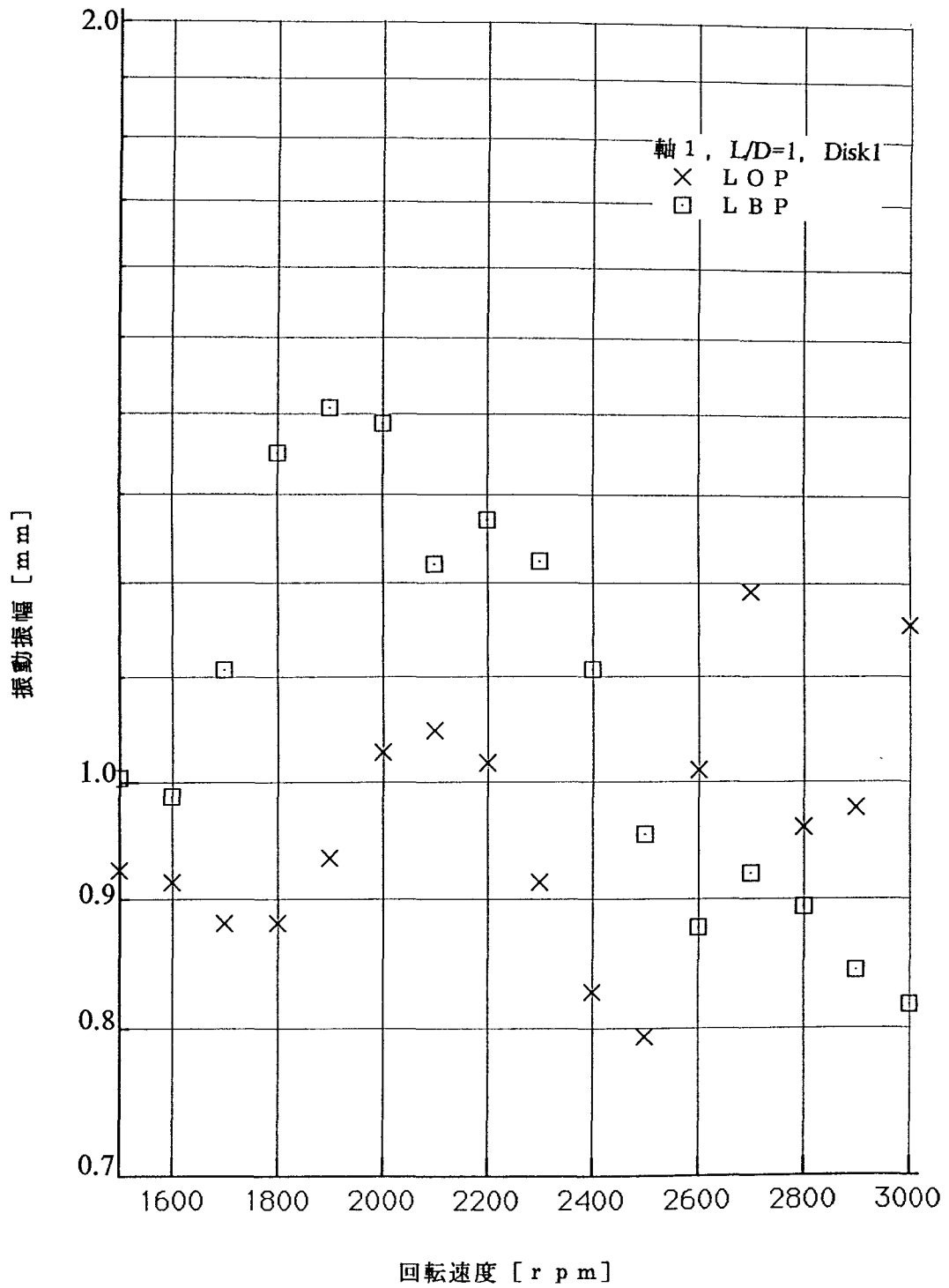


実験結果（本文に載せた結果以外）

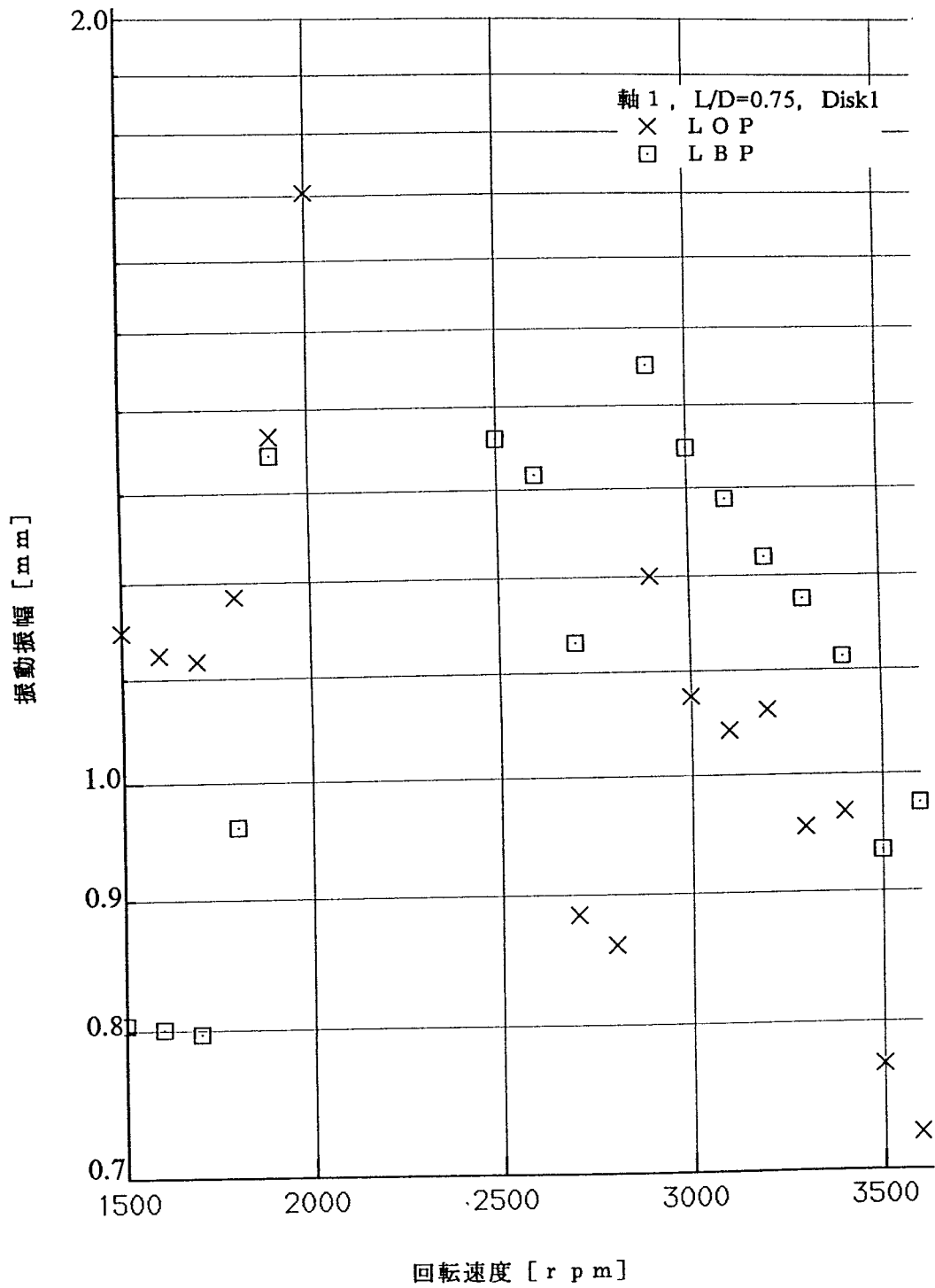


実験結果（本文に載せた結果以外）

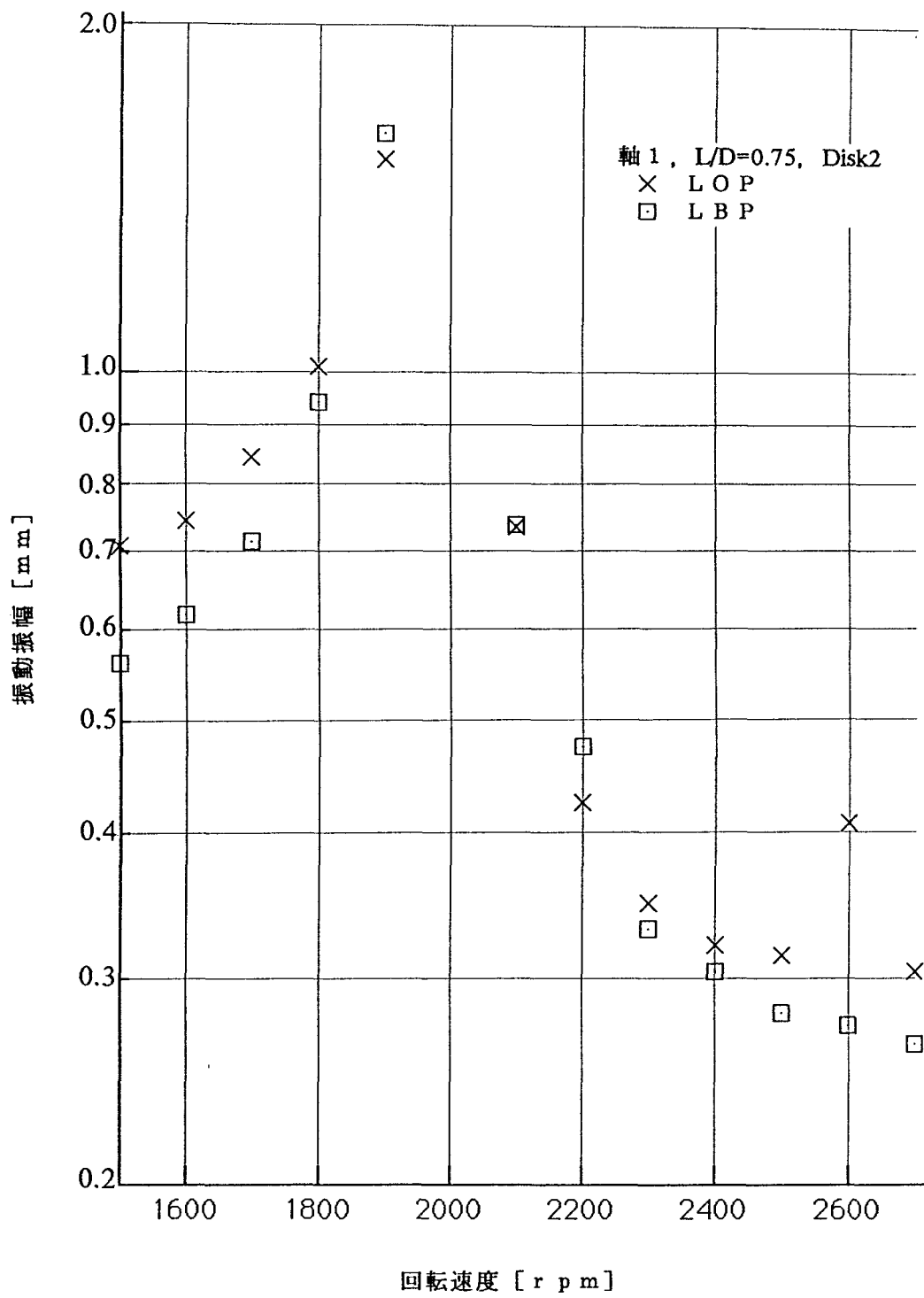




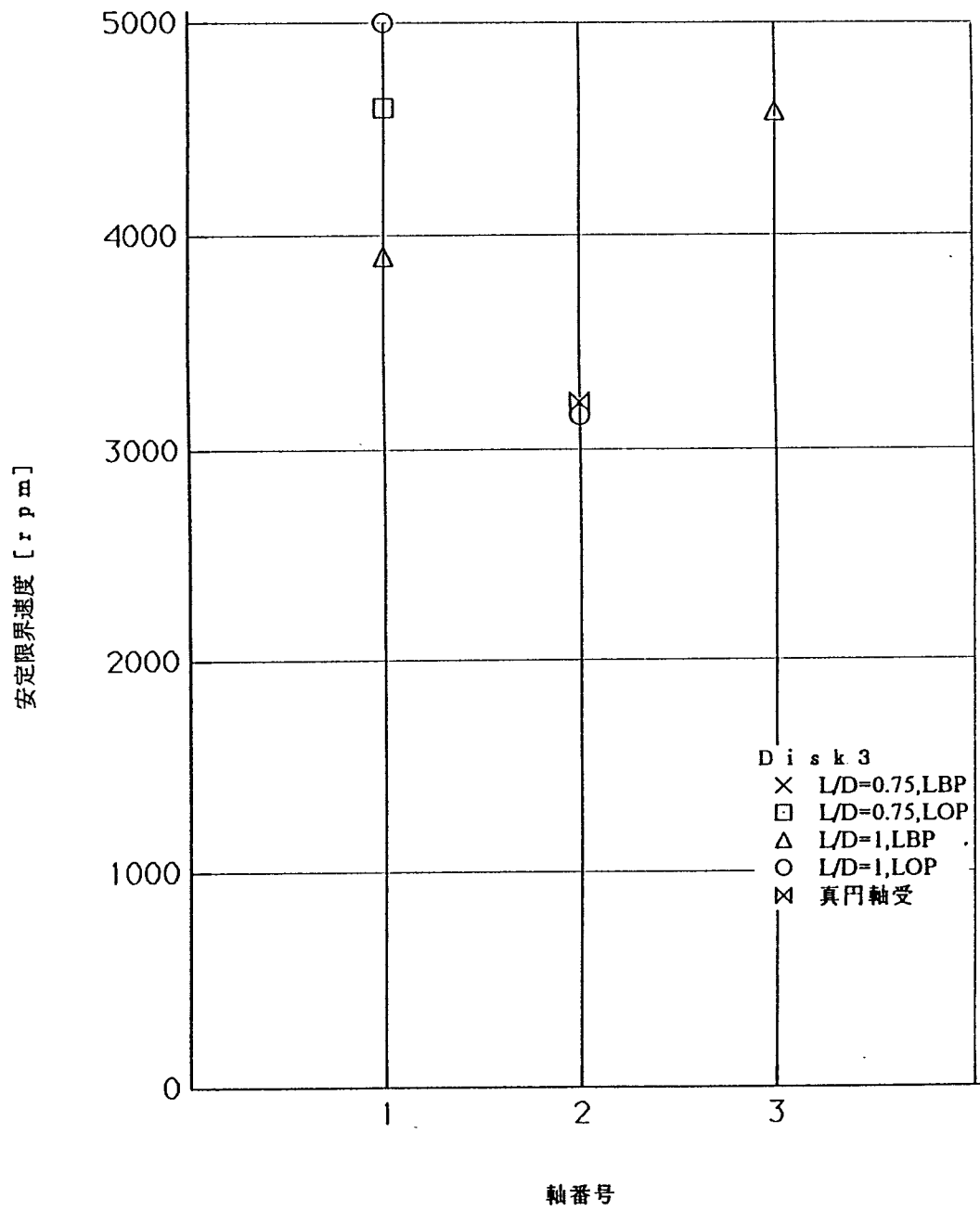
実験結果（本文に載せた結果以外）



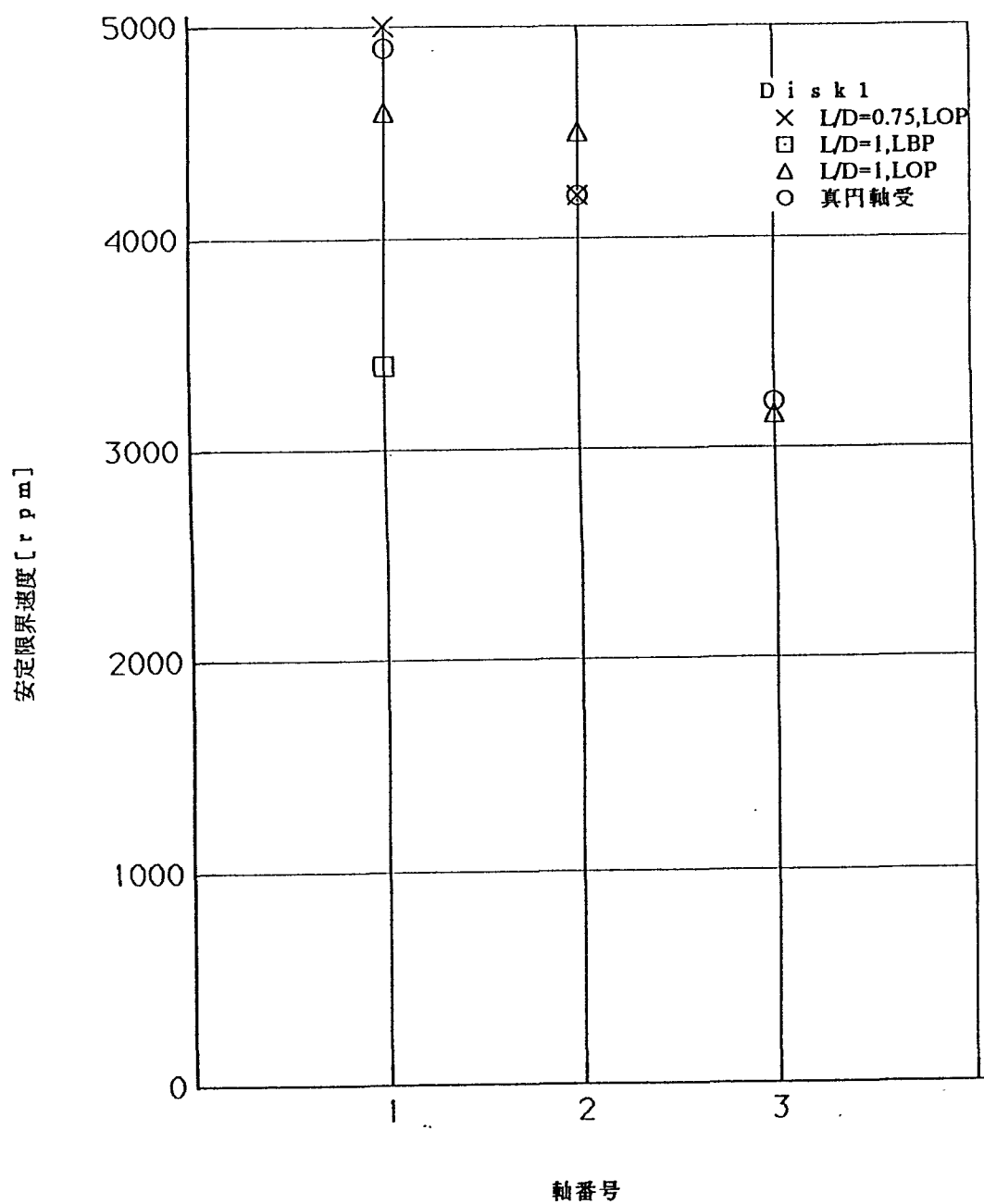
実験結果（本文に載せた結果以外）



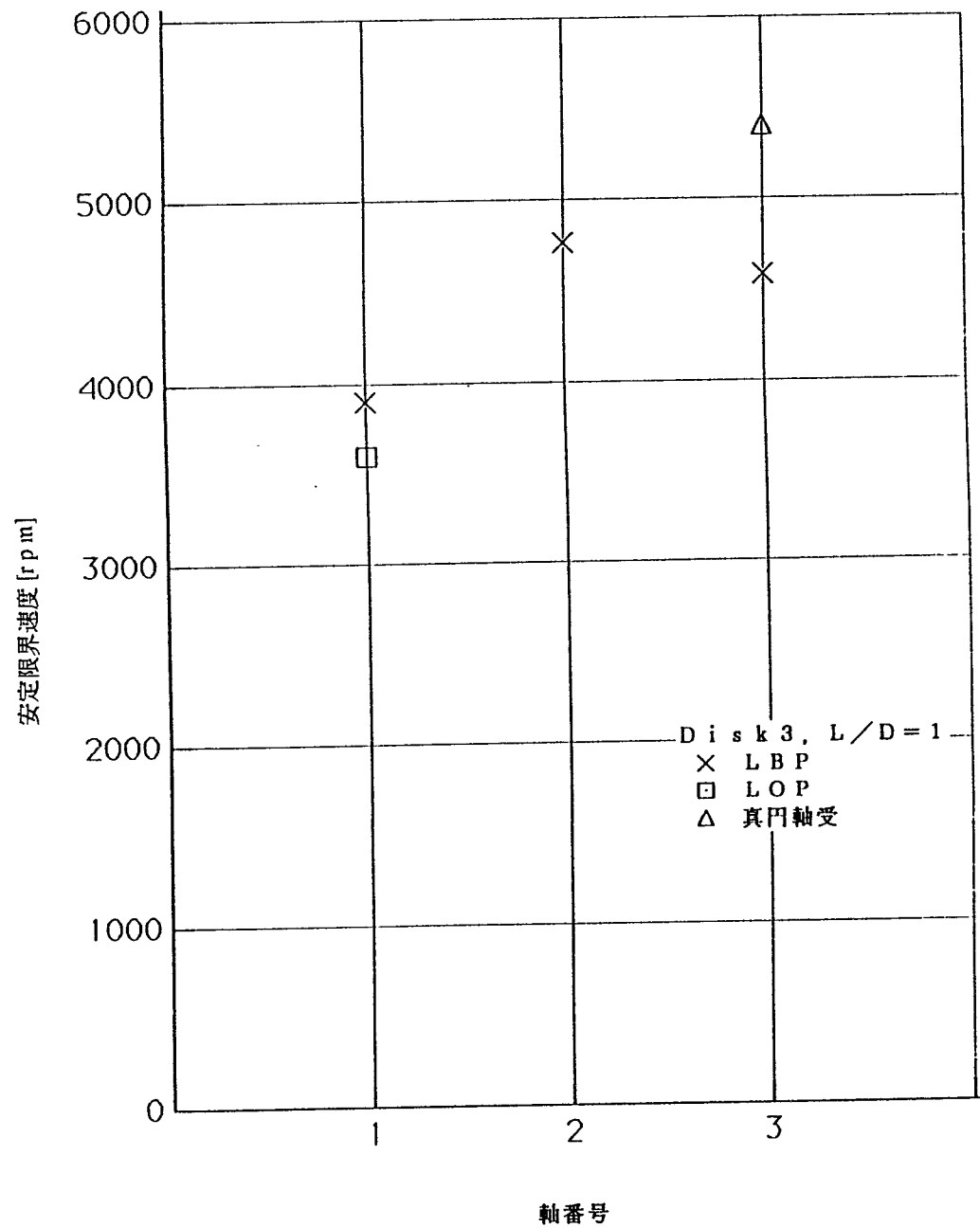
実験結果 (本文に載せた結果以外)



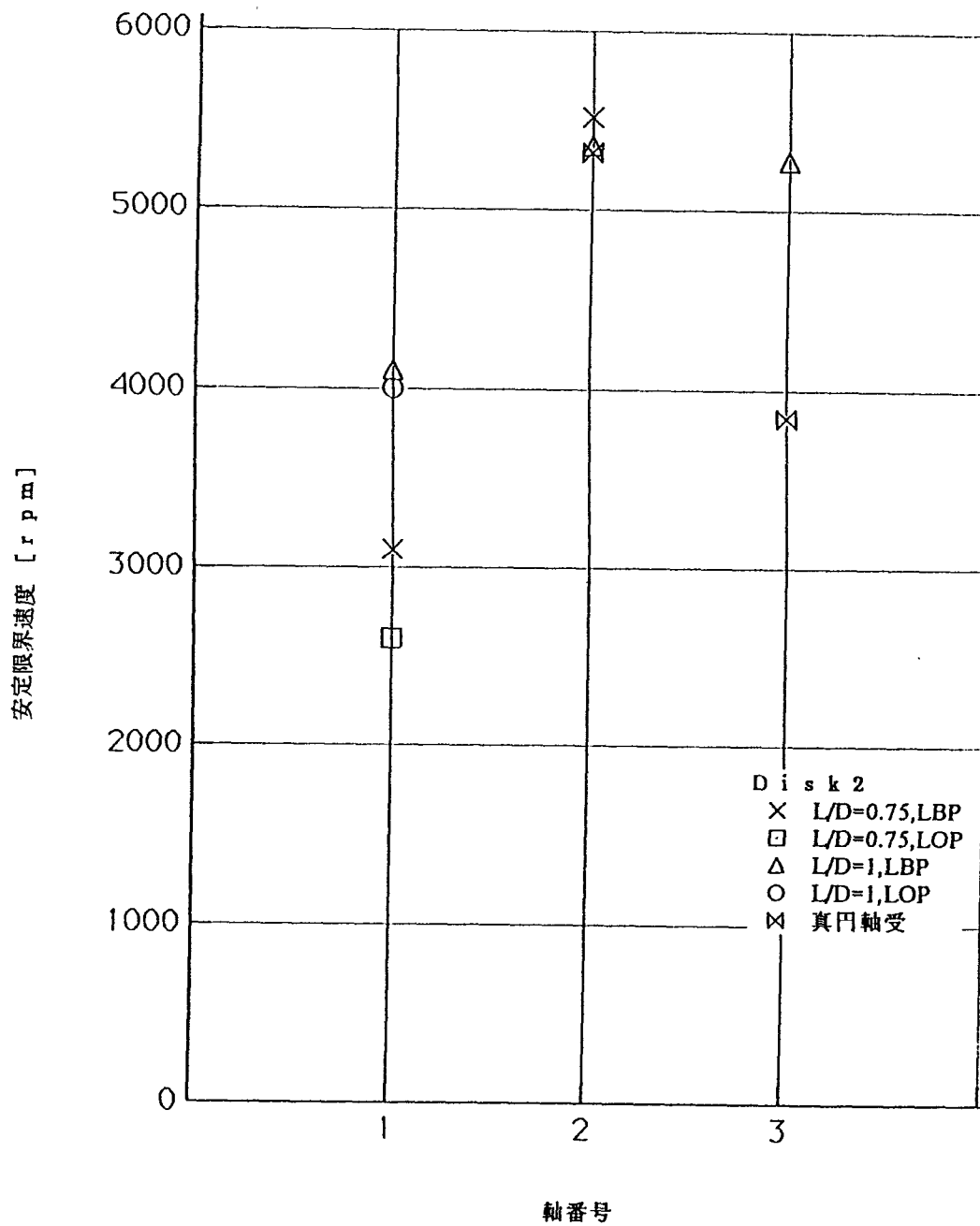
実験結果（本文に載せた結果以外）



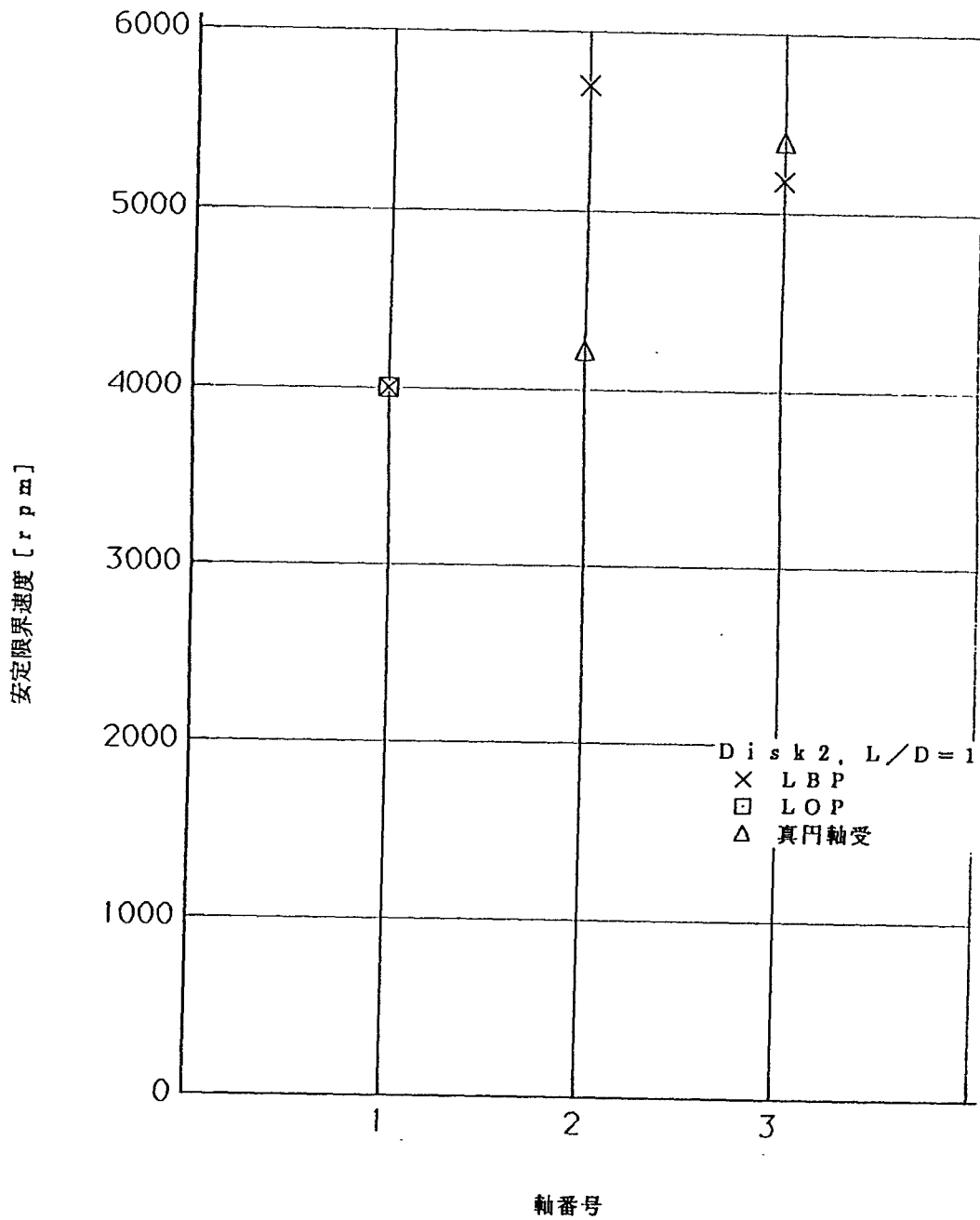
実験結果（本文に載せた結果以外）



実験結果（本文に載せた結果以外）

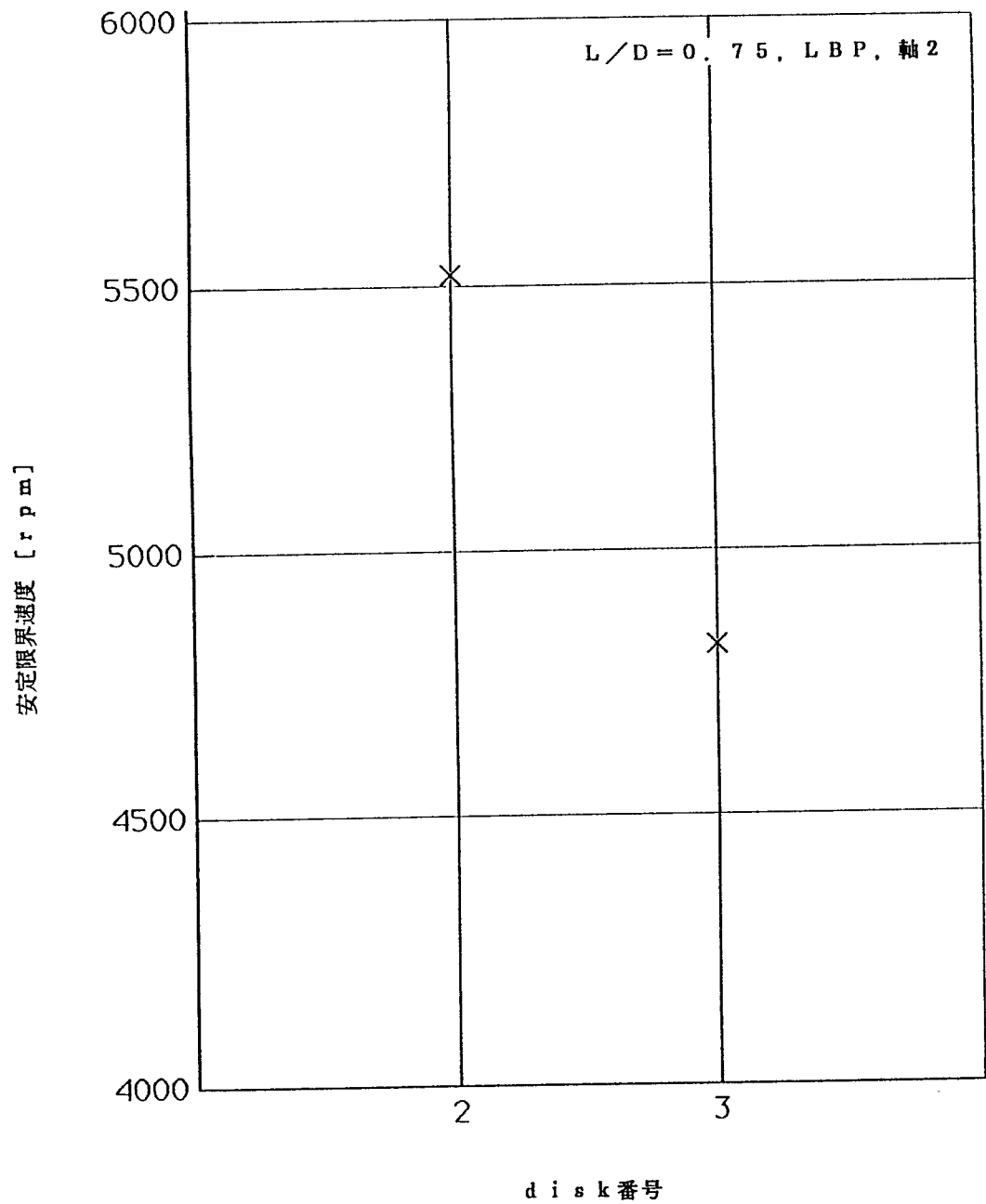


実験結果（本文に載せた結果以外）

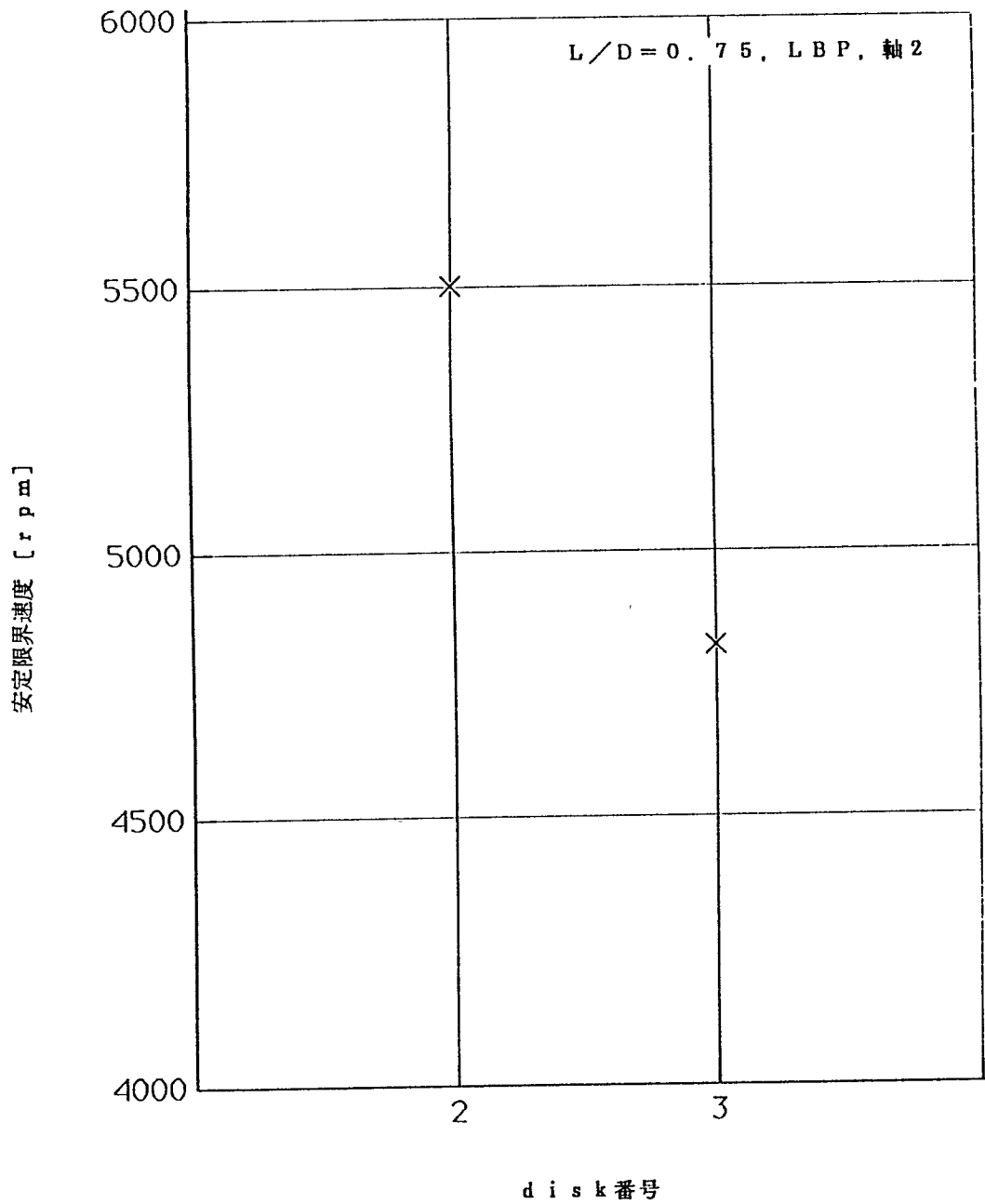


実験結果（本文に載せた結果以外）

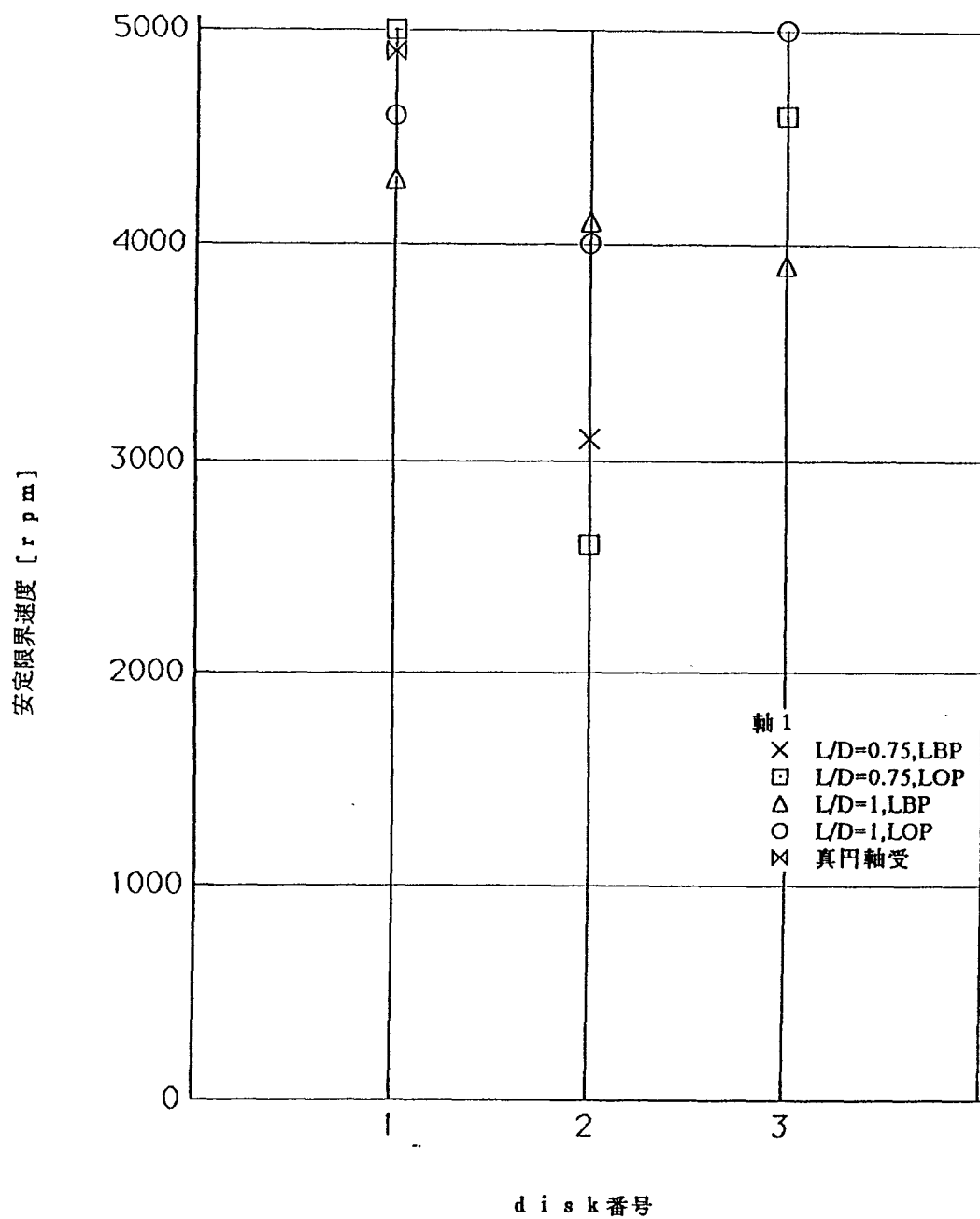




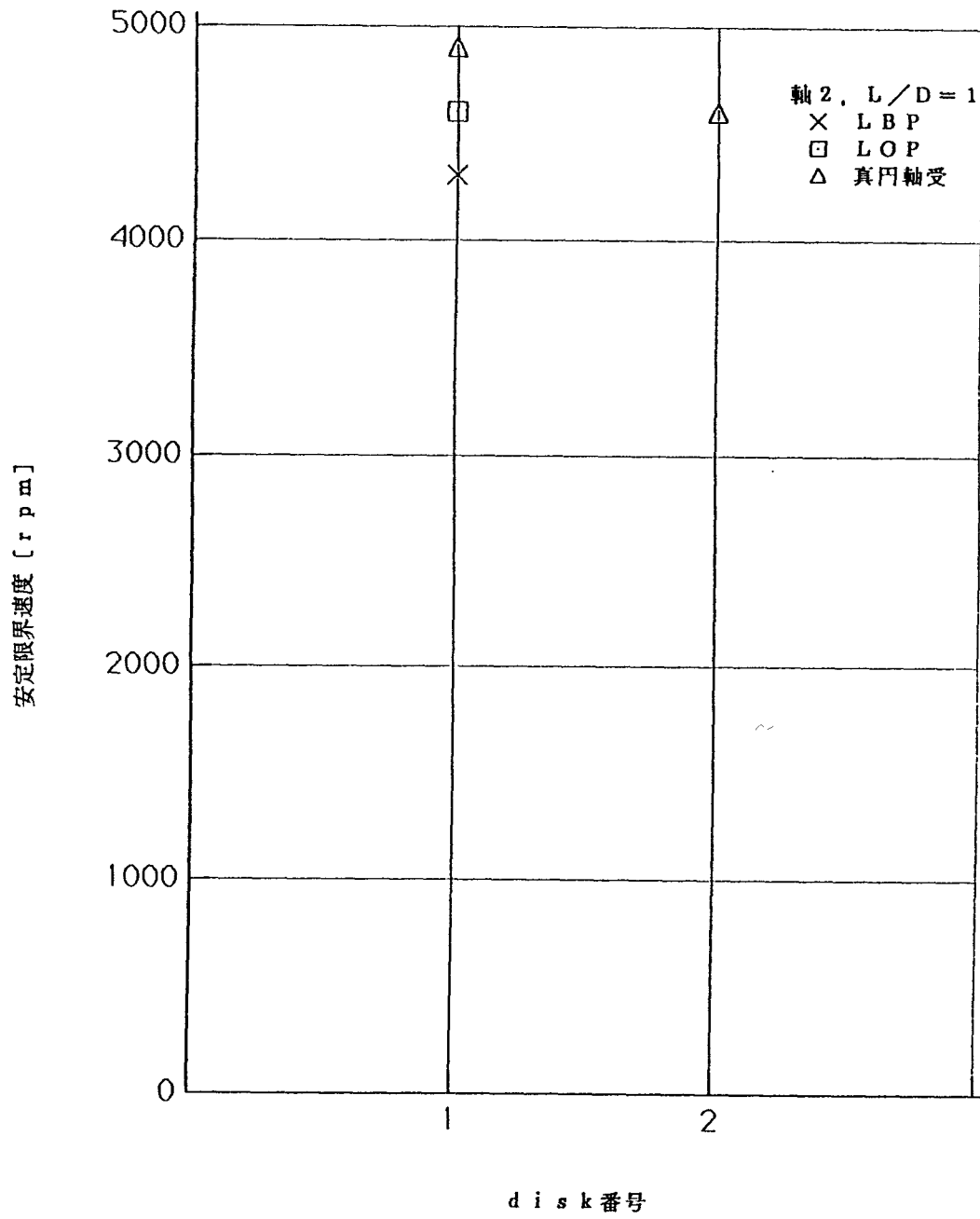
実験結果（本文に載せた結果以外）



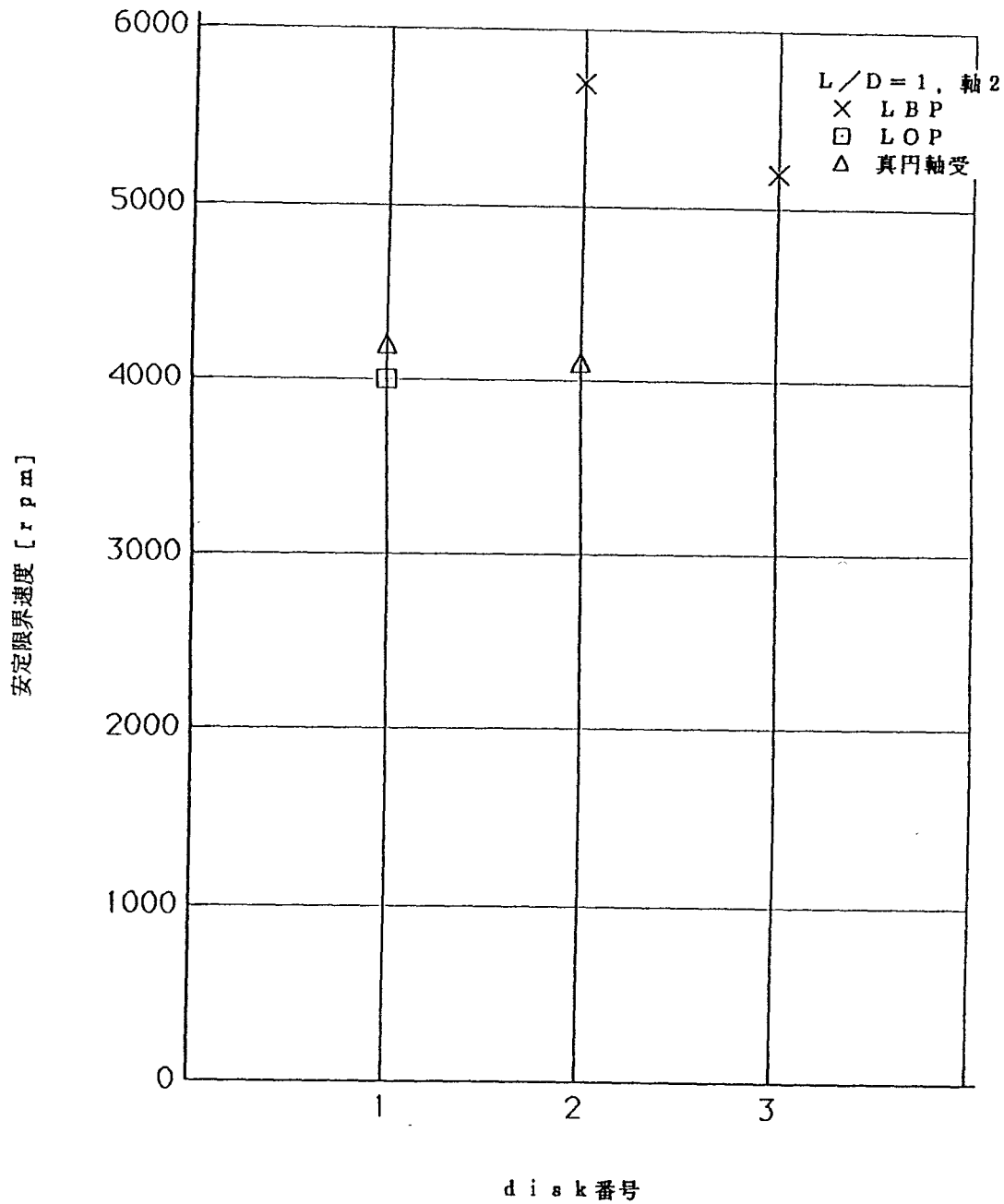
実験結果（本文に載せた結果以外）



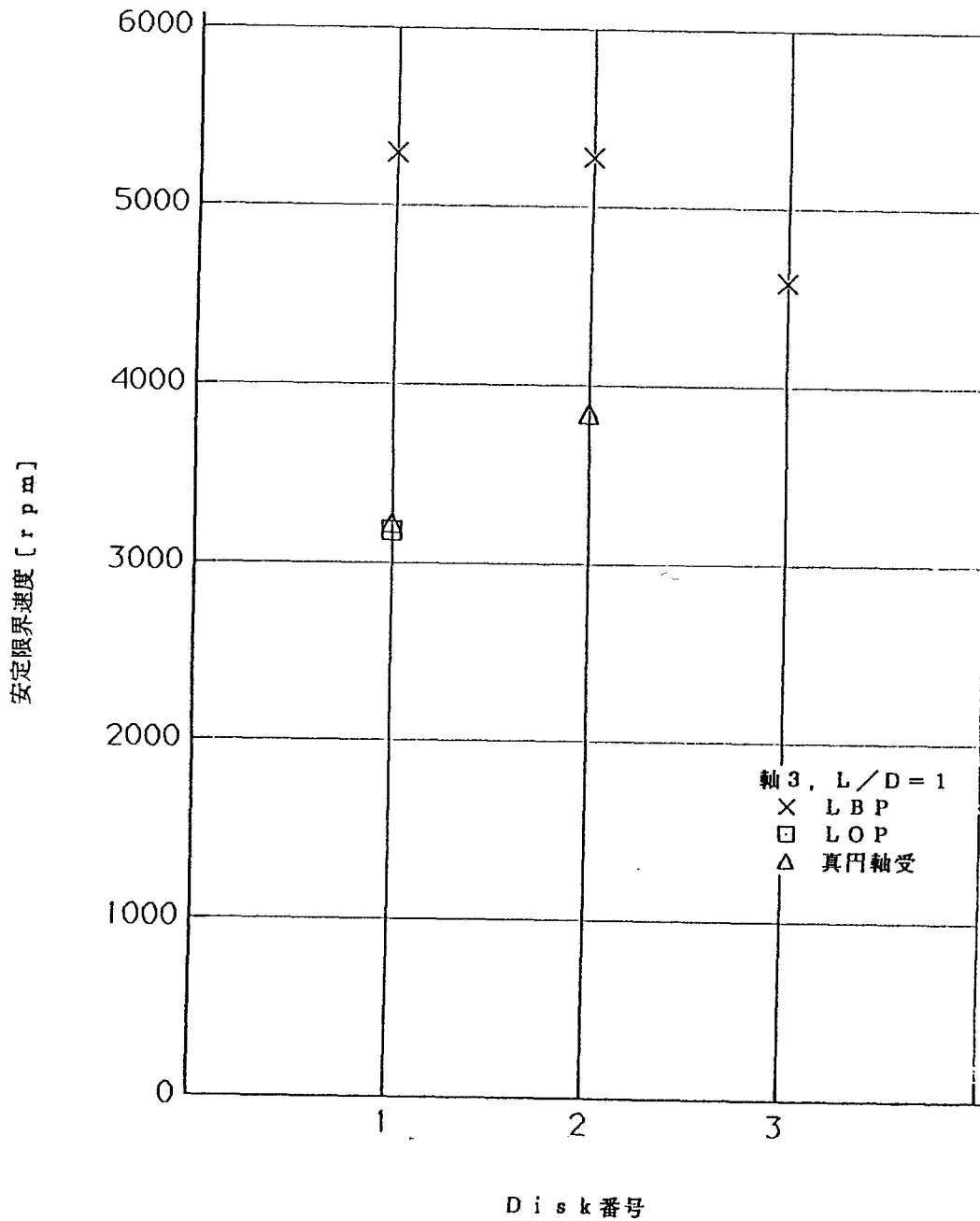
実験結果（本文に載せた結果以外）



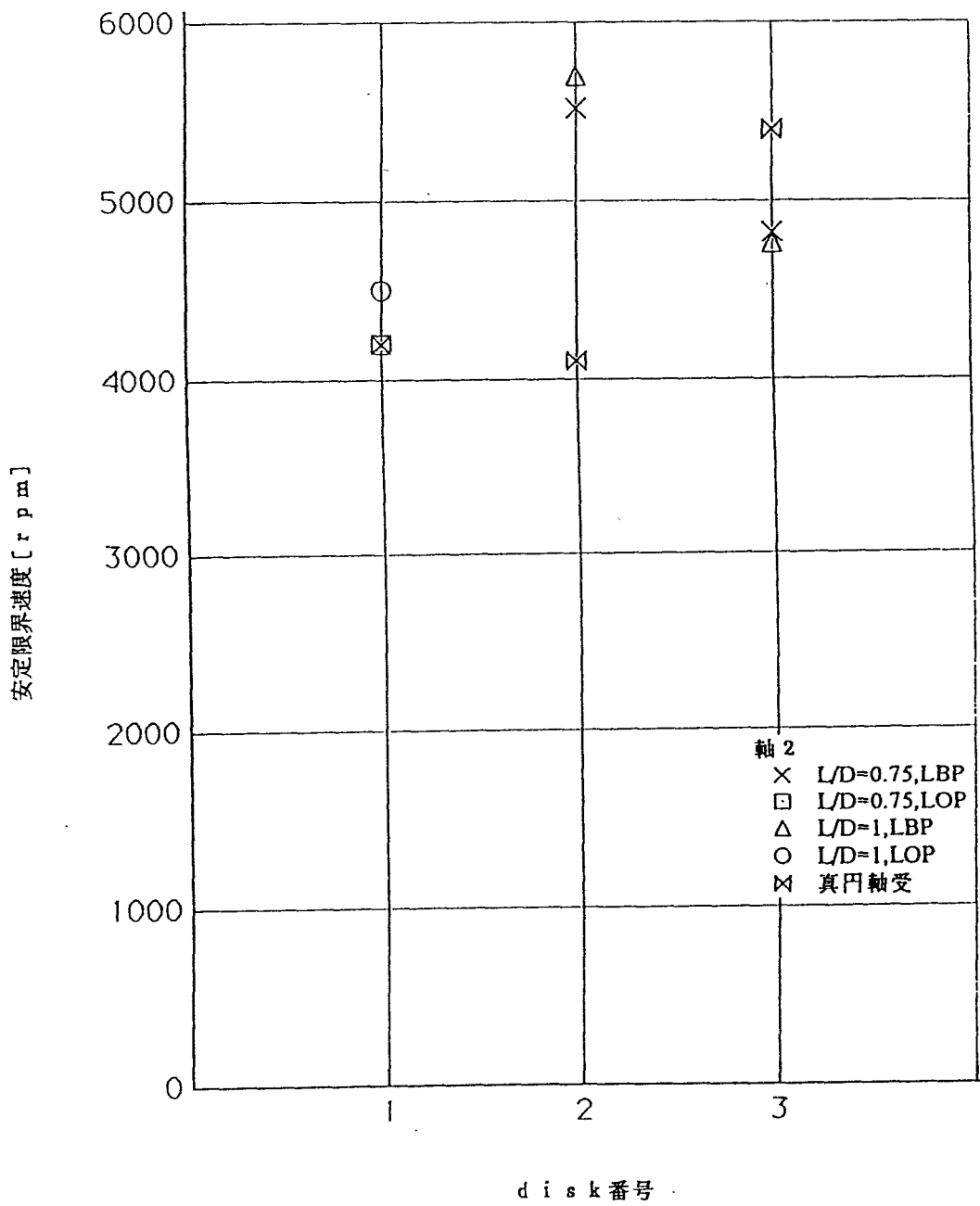
実験結果 (本文に載せた結果以外)



実験結果（本文に載せた結果以外）



実験結果（本文に載せた結果以外）



実験結果（本文に載せた結果以外）